SRI International

Final Report • December 1994

HIGH-PERFORMANCE SPEECH RECOGNITION USING CONSISTENCY MODELING

Vassilios Digalakis, Research Engineer
Hy Murveit, Principal Engineer
Peter Monaco, Research Engineer
Leo Neumeyer, Research Engineer
Ananth Sankar, Research Engineer
Mitch Weintraub, Sr. Research Engineer
Speech Technology and Research Laboratory

SRI Project 3773

Prepared for:

Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA 22217-5000

Attn: Dr. Andre M. van Tilborg
Scientific Officer

Contract N00014-92-C-0154

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

SRI International

Final Report • December 1994

HIGH-PERFORMANCE SPEECH RECOGNITION USING CONSISTENCY MODELING

Vassilios Digalakis, Research Engineer
Hy Murveit, Principal Engineer
Peter Monaco, Research Engineer
Leo Neumeyer, Research Engineer
Ananth Sankar, Research Engineer
Mitch Weintraub, Sr. Research Engineer
Speech Technology and Research Laboratory

SRI Project 3773

Prepared for:

Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA 22217-5000

Attn: Dr. Andre M. van Tilborg
Scientific Officer

Contract N00014-92-C-0154

333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

Final Report • December 1994

HIGH-PERFORMANCE SPEECH RECOGNITION USING CONSISTENCY MODELING

Vassilios Digalakis, Research Engineer
Hy Murveit, Principal Engineer
Peter Monaco, Research Engineer
Leo Neumeyer, Research Engineer
Ananth Sankar, Research Engineer
Mitch Weintraub, Sr. Research Engineer
First Organization

SRI Project 3773

Prepared for:

Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA 22217-5000

Attn: Dr. Andre M. van Tilborg
Scientific Officer

Contract N00014-92-C-0154

Approved:

Patti Price, Director
Speech Technology and Research Laboratory

Donald L. Nielson, Vice President
Computing and Engineering Sciences Division

SRI International 333 Ravenswood Avenue • Menlo Park, CA 94025-3493 • (415) 326-6200 • FAX: (415) 326-5512 • Telex: 334486

1. TECHNICAL SUMMARY

The goal of SRI's consistency modeling project is to improve the raw acoustic modeling component of SRI's DECIPHER[1] speech recognition system and develop *consistency modeling* technology. Consistency modeling aims to reduce the number of improper independence assumptions used in traditional speech recognition algorithms so that the resulting speech recognition hypotheses are more self-consistent and, therefore, more accurate.

At the initial stages of this effort, SRI focused on developing the appropriate base technologies for consistency modeling. We first developed the *Progressive* Search technology that allowed us to perform large-vocabulary continuous speech recognition (LVCSR) experiments. Since its conception and development at SRI, this technique has been adopted by most laboratories, including other ARPA contracting sites, doing research on LVSR.

With an efficient solution for the recognition search problem, we were then able to attack the acoustic modeling problem. An initial attempt to remove independence assumptions from discrete-density hidden Markov model (HMM) based speech recognizers and model dependencies at the intra-segmental level did not provide us with any improvement in accuracy, because of the large number of parameters that it required. To overcome this problem, we developed the *genonic HMM* technology that dramatically reduced the error rate of our speech recognizer and served as a basis for the technology developed in the remainder of the project. The genonic HMM technology was also adopted by other major ARPA contractors in the recent 1994 CSR ARPA benchmarks.

Another goal of the consistency modeling project is to attack difficult modeling problems, when there is a mismatch between the training and testing phases. Such mismatches may include outlier speakers, different microphones and additive noise. We were able to either develop new, or transfer and evaluate existing, technologies that adapted our baseline genonic HMM recognizer to such difficult conditions. These included

- The joint development with ARPA-funded SRI project 4668 of a new speaker-adaptation technique that adapts to the new speaker with only a few minutes of speech. This technique was evaluated in the recent 1994 ARPA benchmarks and can reduce the error rate of nonnative speakers by a factor of 2 to 4 with only five minutes of adaptation speech.
- The transfer of the *probabilistic optimum filtering* (POF) technique developed under NSF funding for microphone and channel independence. The technique was evaluated in the 1993 ARPA benchmarks, where we demonstrated that the recognition performance does not degrade when mismatched microphone types are used in the training and testing phases.

---

1. DECIPHER is a trademark of SRI International.

- The joint development with NSF-funded SRI project 2764 of a technique for recognition in additive noise, which combines the adaptation and POF techniques. The method was evaluated in the recent 1994 ARPA benchmarks, where we found that the error rate, during recognition in additive noise, increases by a factor of 1.6 to 4.9, depending on the signal-to-noise (SNR) ratio. Application of our technique significantly reduces the error rate, and the corresponding increases in error rate after compensation are only 1.2 and 1.8.

SRI emphasizes the development of technology that can be easily transferred into applications. Our genonic HMMs, although very accurate, were significantly more expensive computationally than our previous discrete-HMM technology. In parallel with our work on developing consistency-modeling techniques, a significant effort was undertaken during the second year of the project toward the development of an accurate *real-time LVCSR dictation* system. To achieve this goal, a number of new techniques were developed, and a real-time system that compromised very little accuracy was recently demonstrated to ARPA personnel.

In addition to difficult modeling problems, we have also investigated global consistency modeling, where we try to model dependencies in the speech signal on a scale that is longer than the phone-segment level. We have recently started applying our successful speaker-adaptation techniques to this problem, and we have some preliminary encouraging results. This is an ongoing research effort, and is currently being supported by ARPA-funded SRI project 6429.

Section 2 summarizes the progress made during the project. Papers describing our various techniques are included in the appendix, and are briefly summarized in Section 3.

2. SUMMARY OF PROGRESS

All of our work toward improving the word accuracy of speech recognition systems is evaluated at the yearly ARPA CSR benchmark exercises. The results of these evaluations are summarized in Figure 1 for SRI and three other major ARPA contracting sites. We have plotted the word recognition error rates of each of the four systems for the 1992, 1993, and 1994 'hub' tests. Since the consistency modeling project is primarily focused on acoustic modeling, we present the results for the test conditions that evaluate acoustic modeling technology only and have fixed language modeling and training data across different systems. The results across different years are not comparable, since the tasks and the test sets are different, and the word error rates for year 1992 are plotted for all four sites using twice their actual values. We have chosen to do so solely for display purposes, since the task in 1992 was a 5,000-word dictation task, as opposed to the 20,000-word tasks used in the subsequent years, and it has been empirically found that the 5,000-word error rates are roughly half the 20,000-word ones. Moreover, the results for the November 1994 evaluation are unofficial, since the official National Institute of Standards and Technology (NIST) results were unavailable when this report was written. We can see from this figure the improvement in SRI's performance relative to other sites during the course of the consistency modeling project. At the beginning of the project, SRI's

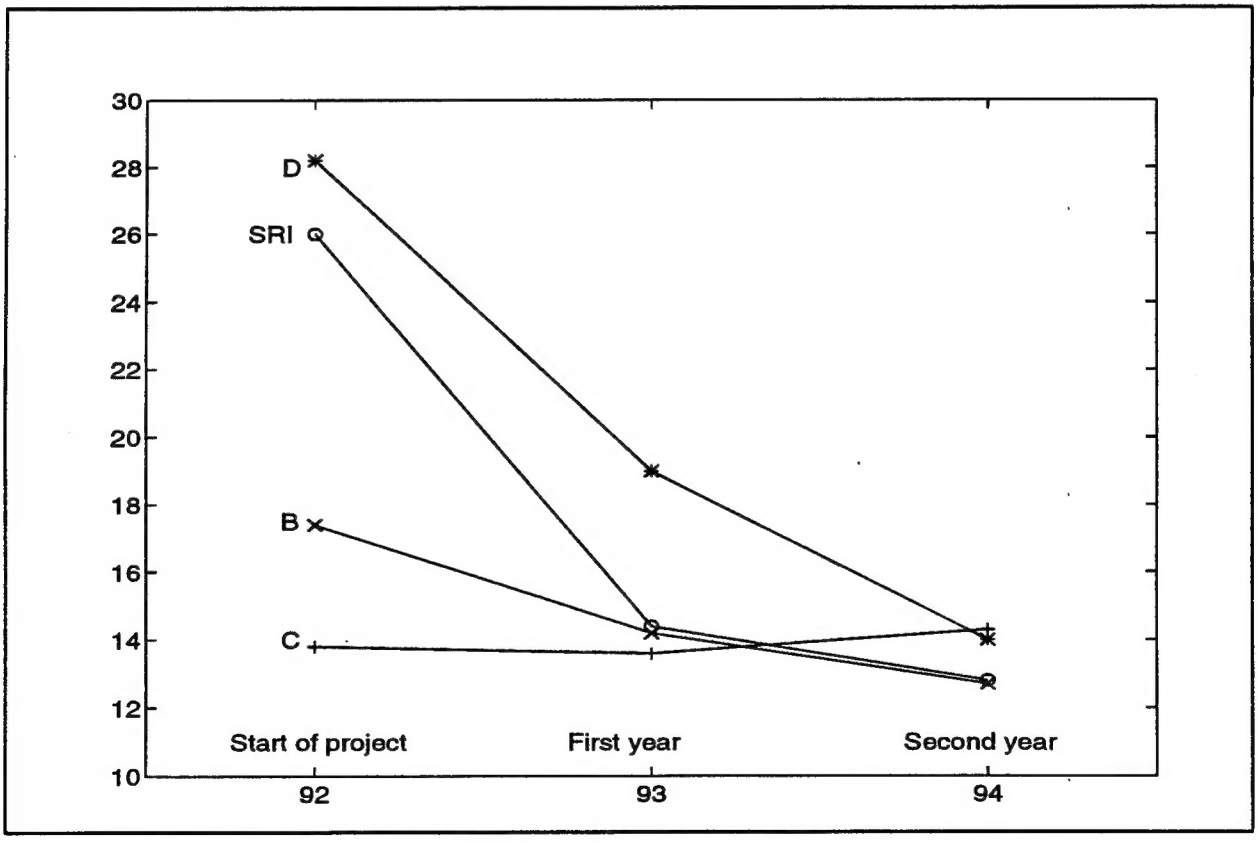

Figure 1. Word-recognition error rate reduction for various ARPA contractors since 1992.

system had twice the error rate of the best-performing system. After two years of project work, and using techniques that were developed in this project and are summarized in the following discussions, SRI's system was one of the best-performing ARPA systems in the 1994 benchmarks.

3. DETAILS OF TECHNICAL DEVELOPMENTS

A number of techniques were either developed or transferred and evaluated using consistency modeling project funding.

3.1 PROGRESSIVE SEARCH TECHNOLOGY

A technique called *Progressive Search,* developed early in the project, allows recognition experiments to be run over several hundred sentences in a few hours instead of a day or more. Progressive Search is a multiple-pass technique, with each pass using a progressively more accurate (and costly) algorithm. Each pass outputs a grammar (word lattice) used to constrain the next pass's search space (instead of a less efficient N-best sentence list). It allows evaluation of computationally demanding algorithms (N-grams, more complex HMMs). It also facilitates developing real-time, high-accuracy, large-vocabulary recognition.

A Progressive Search technique has been applied to a standard cross-word tied-mixture 5K bigram HMM recognizer for ARPA's Wall Street Journal (WSJ) dictation task. It improved recognition development time by an order of magnitude (from 46 x real time to 5.6 x real time) when precomputed first-pass lattices were stored.

Another important application of the Progressive Search technique has been for trigram language models. In this case, the word lattice output by an initial bigram-based recognizer was converted into a trigram word lattice by replicating those states in the lattice where trigram word transition probabilities existed. This approach to trigram language modeling increased decoding time only slightly from that of bigram modeling (15% increase), with a minimal increase to the grammar size (since most of the trigrams were not represented). This approach is much more powerful than using an N-best approach to implementing N-gram language models since more of the correct words exist in the lattice than the top N sentences. For instance, in an experiment using bigram language models for 5,000-word WSJ speech recognition, a system that achieved approximately a 10% word error rate[2] on our development set achieved only an approximately 5% N-best error rate[3] for N = 1000, whereas the relatively compact grammar generated by this system had a 1% lattice error rate.[4] This reduced error rate gives the language model the opportunity to repair errors that the N-best system could not overcome. A paper describing this technique is included in the appendix.

---

2. A 10% bigram error on our development set is roughly equivalent to a 7% word error using bigrams on the official November 1992 evaluation set, approximately the same as the best bigram-based performance reported at the January 1993 ARPA meeting.
3. The N-best word error rate is defined as the average error of the best of the N sentence hypotheses.
4. The lattice error rate is the average of the error rate associated with the best path through the lattices.

3.2 GENONE-BASED HMM TECHNOLOGY

SRI has developed a new type of HMM speech recognition technique called genonic mixtures. A *genone* is a Gaussian codebook used in Gaussian mixture-density HMMs. In this type of system, Gaussian mixture components are shared among groups of states. These groupings are automatically determined using agglomerative clustering techniques. This technique automatically balances the modeling resolution/robustness trade-off, depending on the amount of training data.

We evaluated SRI's DECIPHER technology that existed at the start of this project in ARPA's November 1992 evaluation on the 5,000-word test. This technology was a tied-mixture HMM using SRI's phone set and a combination of SRI's and Dragon's[5] WSJ pronunciation dictionaries. We achieved a 13% error rate. After improving our system between then and June 1993 and using regular progress checks with other development materials (using different speakers than in the November 1992 test set) we reevaluated our speech recognition system on the November 1992 test set[6]. Table 1 shows that improvements made on the choice of phonetic units,

System	Word Error (%)	Sentence Error (%)
SRI, November 1992	13.0	73.9
PTM + cepstral mean removal + phone set + dictionary	9.0	60.6
Genones + above improvements	7.7	53.0

Table 1. Speech recognition accuracy improvement

the dictionary (supplied by the LIMSI laboratory), a cepstral mean removal front end, and, in particular, the use of phonetically tied mixtures (PTM — see appendix) reduced our error rate by 31%[7]. An additional 14% reduction was achieved by the introduction of genone technology, making the overall improvement 41%[8].

The difference in recognition performance between PTM and genonic HMMs was, however, much more dramatic in the WSJ1 portion of the database. There, the training data consisted of 37,000 sentences from 280 speakers, and gender-dependent models were built. The male subset of the 20,000-word, November 1992 evaluation set was used, with a bigram language model. Table 2 compares various degrees of tying by varying the number of genones used in the

5. All product and company names mentioned in this document are the trademarks of their respective holders.
6. The November 1992 test set was used only twice, once in November 1992 and once in June 1993. The particular errors made in November 1992 were not examined; thus, we consider this second test to be a relatively fair evaluation of the progress we made during that period.
7. Our development data experiments suggest that about one half of the 31% improvement is due to PTM.
8. An error rate reduction of 25% was due solely to genone technology.

system. We can see that, because of the larger amount of available training data, the improvement in performance of genonic systems over PTM systems is much larger (20%) than in our 5,000-word experiments. Moreover, the best performance is achieved for a larger number of genones—1,700 instead of the 495 used in the 5,000-word experiments. A paper describing this technique is included in the appendix.

	PTM	Genonic HMMs			
Number of Genones	40	760	1250	1700	2400
Word error rate (%)	14.7	12.3	11.8	11.4	12.0

Table 2. Recognition performance on the male subset of the 20,000-word WSJ November 1992 ARPA evaluation set for various numbers of codebooks using a bigram language model

3.3 LOCAL CONSISTENCY MODELING

Local consistency modeling attempts to remove the independence assumption of nearby frames, but not frames across the entire input sentence. The spectral input to the HMM system at neighboring frames is highly correlated because (1) the speech signal is sampled faster (every 10 ms) than the vocal tract changes, and (2) the spectral analysis between neighboring frames uses overlapping windows (25.6 ms). Therefore, the HMM independence assumption is clearly violated, and recognition performance could improve by modifying the HMM model to capture this correlation between neighboring frames. In addition, sources of variability such as microphone, vocal tract shape, speaker dialect and speech rate will not dominate the likelihood computation during Viterbi decoding by being rescored at every frame.

3.3.1 Discrete Density HMMs

The time correlation can be modeled by replacing the standard output distribution $p(x_t \mid s)$ of the observed spectral feature x_t given the HMM state s with a model that can account for the previous acoustic history $p(x_t \mid H_t, s)$, where H_t is the summary of the previous acoustic input. A straightforward implementation is to represent the summary of the previous acoustics H_t by x_{t-1}: the current frame is highly correlated with the previous acoustic frame, and although prediction of the current frame using a longer history and spectral dynamics is theoretically better, a good first-order approximation that uses only the last observation may be sufficient. This approach does not introduce any significant problems in an HMM-based recognizer, since the output-independence assumption is not necessary for the development of the HMM recognition (Viterbi) and training (Baum-Welch) algorithms. Both of these algorithms can be modified to cover the case when the features depend not only on the current HMM state, but also on features at previous frames [Wellekens87].

In a discrete density system, we can use the state conditional output probabilities $p(q_t \mid q_{t-1}, s)$ where q_t is the vector-quantized speech signal at time t. In a typical large-vocabulary speech recognizer such modeling would require the estimation of a very large number

of parameters ((number of states = 10,000) x (codebook size=256)^2 = 650 million parameters per feature). To reduce the number of parameters, we made certain simplifying assumptions that reduced the number of parameters that need to be estimated to 12 million parameters per feature, and we then tested this approach on one of our WSJ development test sets. The results are summarized in Table 3.

Word Error for System	Standard Recognizer	Recognizer with Co-Occurrence Local Consistency
Context-Independent Models	46.2	41.8
Context-Dependent Models	20.7	22.0

Table 3. Word error for WSJ male 5,000-word closed verbalized punctuation development test

While the context-independent model results improved, the context-dependent model performance decreased. We attributed this result to the large number of parameters that needed to be estimated. The number of parameters increases proportionately with the square of the codebook size. It is, therefore, essential to decrease the codebook size, and this can be achieved by using continuous-density genonic HMM systems.

3.3.2 Continuous-Density HMMs

To overcome the parameter-estimation problem associated with modeling correlations using discrete-density HMMs, we focused on modeling time correlation using the continuous-density genonic HMMs described in Section 3.2. With the exception of the work reported in [Digalakis93] that was based on segment models, explicit time-correlation modeling has not improved the performance of HMM-based speech recognizers [Brown87, Kenny90]. To investigate these results, SRI conducted a study to estimate the potential improvement in recognition performance when using explicit correlation modeling over more traditional methods like time-derivative information. We used information-theoretic criteria and measured the amount of mutual information between the current HMM state and the cepstral coefficients at a previous "history" frame. The mutual information was always conditioned on the identity of the left phone, and was measured under three different conditions:

- *I(h,s)*—unconditional mutual information between the current HMM state and a cepstral coefficient at the history frame; a single, left-context-dependent Gaussian distribution for the cepstral coefficient at the history frame was hypothesized.
- *I(h,s|c)*—conditional mutual information between the current HMM state and a cepstral coefficient at the history frame when the same cepstral coefficient of the current frame is given; a left-context-dependent, joint Gaussian distribution for the cepstral coefficients at the current and the history frames was hypothesized.
- *I(h,s|c,d)*—the same as explained above, but conditioned on both the cepstral coefficient and its corresponding derivative at the current frame.

The results are summarized in Table 4 for history frames with lags of 1, 2, 4 and a variable one. In the latter case, we condition the mutual information on features extracted at the last frame of the previous HMM state, as located by a forced Viterbi alignment. We can see from this table that in the unconditional case, the cepstral coefficients at frames closer to the current one provide more information about the identity of the current phone. However, the amount of additional information that these coefficients provide when the knowledge of the current cepstra and their derivatives is taken into account is smaller. In addition, the additional information in this case is larger for lags greater than 1, and is maximum for the variable lag.

Information Lag d	0	1	2	4	Variable
$I(h, s)$	0.28	0.27	0.25	0.19	0.25
$I(h, s \mid c)$	0	0.13	0.15	0.15	0.21
$I(h, s \mid c, d)$	0	0.11	0.14	0.13	0.20

Table 4. Mutual information (in bits) between HMM state s at time t and cepstral coefficient h at time $t-d$ for various lags. Included is the conditional mutual information when the corresponding cepstral coefficient and its derivative at time t are given.

This would predict that the previous frame's observation is not the optimal frame to use when conditioning a state's output distribution. To verify this, and to actually evaluate recognition performance, we incorporated time-correlation modeling in SRI's most accurate recognition system that uses genonic mixtures. We found that the recognition results were in perfect agreement with the behavior predicted by the mutual-information study. The improvements in recognition performance for fixed-lag history frames over the system that does not use conditional distributions were moderate and proportional to the measured amount of conditional mutual information at these frames. This information is currently captured in SRI's speech recognizer through the use of linear discriminant analysis, as explained in [Digalakis94] and the results are summarized in Table 5.

System	Bigram LM	Trigram LM
Baseline Genonic HMM	20.5	17.0
Genonic HMM + Linear Discriminants	19.1	15.8

Table 5. Word error rates (%) on the 20,000-word open-vocabulary male development set of the WSJ1 corpus with and without linear discriminant transformations

According to the mutual information results, we should expect a significant improvement in recognition performance when modeling the dependencies between the current frame and the last frame of the previous state — that is, when we model the dynamics across the whole subphonetic segment and condition the output HMM distributions not only on the previous output frames, but also on the segment start time. The observation that the incorporation of segmental

features and modeling of the segment dynamics can improve recognition performance is consistent with previous results by other researchers [Ostendorf89] [Digalakis93] [Algazi93], and we are currently doing research in this area.

3.4 REAL-TIME WALL-STREET JOURNAL DICTATION SYSTEM

A significant effort in this project was invested in the development of a real-time continuous-density dictation system. While genone-based systems perform very well, they are computationally costly, as a large number of Gaussians need to be computed during recognition. We researched a variety of ideas to reduce the recognition time and created a real-time continuous-density dictation system while maintaining good recognition performance.

3.4.1 Efficient Computation of Gaussian Probabilities

Genonic HMM recognition systems require evaluation of very large numbers of Gaussian distributions, and can be very slow during recognition. The baseline system referenced here uses 589 genonic mixtures (genones), each with 48 Gaussian distributions, for a total of 28,272 39-dimensional Gaussians. On ARPA's November 1992 20,000-word evaluation test set, this noncrossword, bigram system performs at a 13.43% word error rate. Decoding time from word lattices is 12.2 times slower than real time on an R4400 processor. Full grammar decoding time would be much slower. Since the decoding time of a genonic recognition system such as this one is dominated by Gaussian evaluation, one major thrust of our effort to achieve real-time recognition has been to reduce the number of Gaussians requiring evaluation each frame. We briefly describe two innovations we have used to reduce the number of Gaussians that need to be computed. For details, a paper describing these techniques is included in the appendix.

Gaussian clustering

We may reduce the total number of Gaussians per genone by clustering groups of Gaussians to form a single Gaussian density. Specifically, we used an agglomerative procedure to cluster the component densities within each genone to a smaller number. To do this, we must define a distance metric between Gaussians. We considered several criteria that were used in [Kannan94], such as entropy-based and generalized likelihood-based distortion measures. We found that the entropy-based measure worked better. Specifically, the cost of pooling two densities is the increase in entropy due to this pooling. Once we have reduced the number of Gaussians by clustering, we can reestimate their parameters using the Baum-Welch algorithm. This reestimation ensures state observation densities are better represented.

In Table 6, we show the word error rate as a function of the number of Gaussians per genone. In the first row, we show the baseline case of 48 Gaussians per genone. When these are reduced to 18 Gaussians per genone using clustering, it is seen that the word error rate increases slightly (from 13.4% to 14.2%). However, by using just one iteration of the Baum-Welch algorithm to reestimate these Gaussians, the error rate becomes 13.6%, which is as good as the baseline error rate. In addition, we see that clustering, and then reestimating the Gaussians is superior to estimating a smaller number of Gaussians from scratch (last row of the table). The

table shows that we can reduce the number of Gaussians by about a factor of 3 (from 48 to 18), while maintaining the error rate.

System	Gaussians per Genone	Word Error (%)
Baseline1	48	13.43
Baseline1+Clustering	18	14.17
above+Retraining	18	13.64
Baseline2	25	14.35

Table 6. Improved training of systems with fewer Gaussians by clustering from a larger number of Gaussians

Gaussian shortlists

Although clustering significantly reduces the total number of Gaussians, all the Gaussians belonging to genones used by HMM states that are in the Viterbi beam search must be evaluated at each frame during recognition. This evaluation includes a large amount of unnecessary computation; we have verified experimentally that the majority of the Gaussians will yield negligible probabilities. As a result, after reducing the Gaussians by a factor of 3 using clustering, the decoding time from word lattices is still 7.9 times slower than real time.

We have developed a method similar to the one introduced by Bocchieri [Bocchieri93] for preventing a large number of unnecessary Gaussian computations. Our method is to partition the acoustic space and for each partition to build a *Gaussian shortlist* — a list specifying the subset of the Gaussian distributions expected to have high likelihood values in a given region of the acoustic space. First, vector quantization (VQ) is used to subdivide the acoustic space into regions. Then, one list of Gaussians is created for each combination of VQ region and genone. The lists are created empirically, by considering a sufficiently large amount of speech data. For each acoustic observation, each Gaussian distribution is evaluated. Those distributions whose likelihoods are within a predetermined fraction of the most likely Gaussian are added to the list for that VQ region and genone.

When recognizing speech, each observation is vector quantized, and only those Gaussians found in the shortlist are evaluated. When this technique was used on the clustered genonic system described earlier, it resulted in shortlists with an average of 2.48 Gaussians per genone with no degradation in recognition accuracy.

Figure 2 summarizes our results on computational reduction for Gaussian evaluations during recognition. We started with a speech recognition system with 48 Gaussians per genone (a total of 28,272 Gaussian distributions) that evaluated 14,538 Gaussian likelihood scores per frame and achieved a 13.4% word error rate running 12.2 times slower than real time on word lattices. Combining the clustering and Gaussian shortlist techniques, we decreased the average number of Gaussians contained in each list to 2.48. As a result, the system's computational requirements

were reduced to 732 Gaussian evaluations per frame, resulting in a system with word error of 13.5% (identical to the baseline system), running at 2.5 times real time from word lattices.

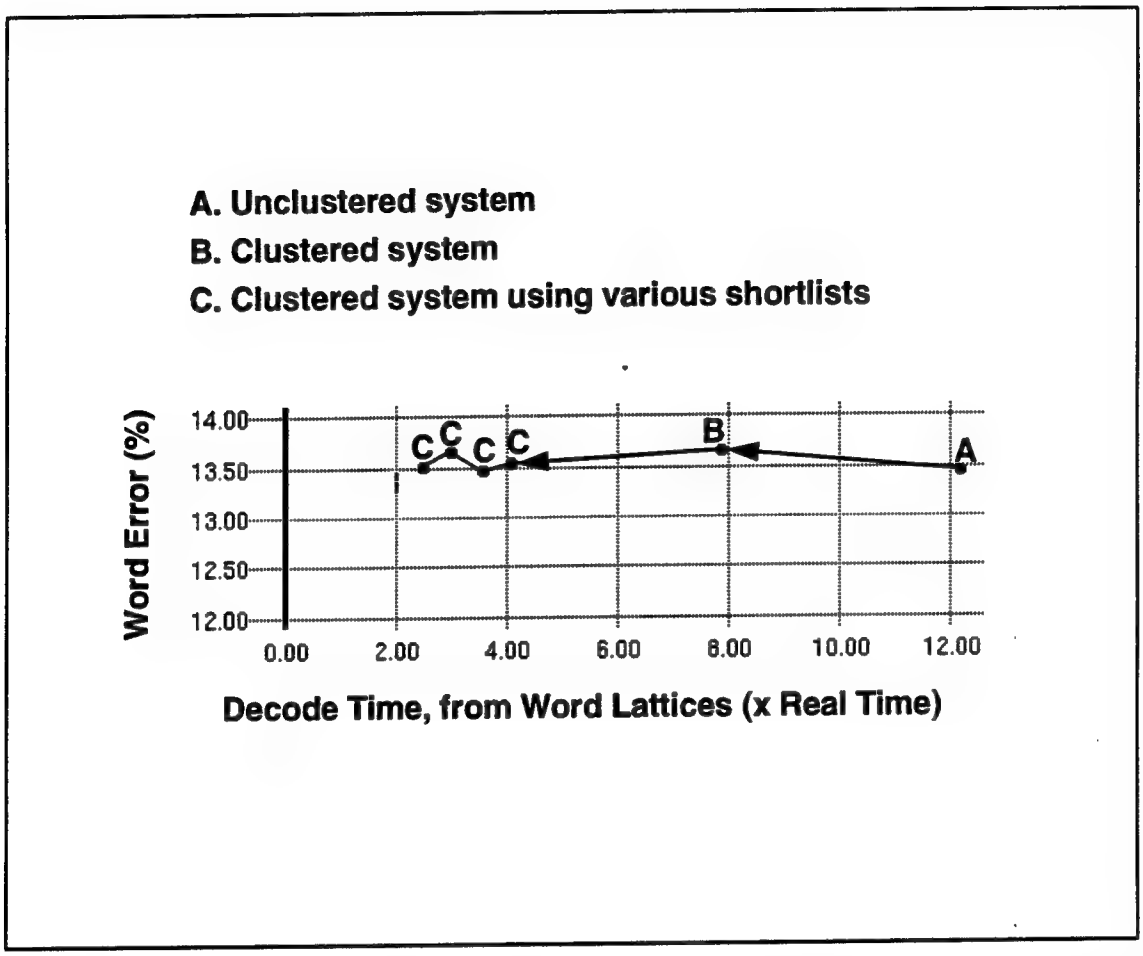

Figure 2. Word error rate as a function of the decoding time for the baseline system (A) and systems with fast Gaussian evaluation schemes (B and C).

3.4.2 Efficient Grammar Organization

There is an intrinsic trade-off between recognition accuracy and recognition time. It is necessary to realize an attractive trade-off for both a real-time system and one that can be useful for research. Traditional approaches include adjusting the beamwidth of the Viterbi search and using less costly output distribution models, such as discrete density HMMs. We have explored some efficient grammar organization techniques to realize a good trade-off between speed and recognition accuracy.

Lexical trees

We explored the use of lexicon trees as a technique for speeding up the recognition process. Lexicon trees represent the phonetics of the recognition vocabulary as a tree instead of as a list of pronunciations (lists of phones). With a tree representation, words starting with the same phonetic units share the computation of phonetic models. Because of the large amount of sharing, trees can drastically reduce the amount of computation required by a speech recognition system.

There are, however, some drawbacks to this approach. First, triphone modeling is affected because of possible ambiguities in the right context of triphones in the lexical tree. Second, we cannot use bigram probabilities to prune word hypotheses before computing the word acoustic probabilities because the identity of the word being decoded is not known until the search process reaches a leaf of the tree. The first problem is not so serious since a large number of triphones in the tree have unambiguous right contexts. However, the problem can be handled by replicating triphones with different right contexts in the lexical tree. We have addressed the second problem by a method called Approximate Bigram Trees. In an approximate bigram tree, the aim is to model the salient portion of the backed-off bigram language model [Katz87] in use. In an approximate bigram tree, a standard lexicon tree (incorporating unigram word probabilities) is combined with a bigram section that maintains a linear (nontree) representation of the vocabulary. Bigram and backoff language model transitions are added to the leaves of the tree and to the word-final nodes of the bigram section. When the entire set of the bigram is represented, this network implements a full backed-off bigram language model with an efficient tree-based backoff section. In fact, for VQHMM systems, this scheme halves our typical decoding time for little or no cost in accuracy. Typically, however, we need further reduction in the computational requirement. To achieve this we represent only a subset of the group of bigram transitions (and adjust the backoff probabilities appropriately). This degrades the accuracy of our original bigram language model, but reduces its computational requirements. The choice of which bigrams to represent is the key design decision for approximate bigram trees. We have experimented with various techniques for choosing bigram subsets as detailed in [Murveit94].

After-the-fact language model

Another method of incorporating a language model in a lexical tree is to use after-the fact language modeling. In this scheme, at every frame the list of word endings and their probabilities are stored. An acoustic search is then carried out using a lexical tree. At the end of the tree, each ending word is backtraced to locate the best predecessor word, and the corresponding bigram probability is applied. We have also used this scheme to apply trigram probabilities. The after-the-fact language model is faster than the approximate bigram trees, at the cost of suboptimal recognition accuracy.

3.4.3 Multiprocessing

We used multiprocessing to speed up the recognition process. We implemented this on a four-processor SPARC 20. One processor was used to perform the search, while four processors were used to compute the Gaussian probabilities.

3.4.4 Results

As a result of the ideas we have described, we were able to develop a real-time continuous-density 20,000-word dictation system using trigram language models. The recognition accuracy of this system was 84% as compared to 91% for a non-real-time system where trigram rescoring is used. The significant innovations used to achieve this result were the Gaussian clustering and shortlists to decrease the number of Gaussians computed, a more efficient organization of the grammar using lexical trees and after-the-fact language modeling, and multiprocessing to share the computational load among different CPUs.

3.5 RECOGNITION IN MISMATCHED CHANNELS

Under NSF funding, SRI developed a technique called *Probabilistic Optimum Filtering* (POF) that allows a recognizer to operate in adverse acoustic environments. A mapping is established between the clean acoustic space, used to estimate the HMMs, and the noisy space, in which the recognizer has to operate. To estimate the mapping parameters, a small stereo database with simultaneous recordings of the clean and noisy spaces must be available.

The POF technique has been applied to the problem of microphone mismatch between the training and testing phases, where the goal is to be able to use the same recognizer with a variety of microphones and channels, so that the recognizer does not need to be retrained for each new acoustic environment. In the past, we have experimentally tested the mapping on over-the-telephone recordings. We found that using POF with wideband HMMs results in higher recognition accuracy than using narrowband telephone models directly.

Using consistency modeling project funding, we transferred the POF technique to the WSJ domain, and tested the algorithm on data recorded with a desktop microphone in the November 1993 ARPA benchmarks. The results showed that after compensating for the mismatch between the close-talk and the desktop microphones, the recognition performance of the secondary microphone is almost as good as on the one used to train the models. A paper about the 1993 SRI Spoke evaluation is included in the appendix.

3.6 SPEAKER ADAPTATION

Automatic speech recognition performance degrades rapidly when there is a mismatch between the testing and the training conditions under which the recognizer parameters were estimated. It may not always be feasible to have consistent conditions in the testing and training phases. For example, in large-vocabulary dictation applications the speaker-independent performance degrades dramatically for outlier speakers, such as nonnative speakers of the recognizer language. Since modern large-vocabulary speech recognizers have millions of free parameters, it is not practical to collect large amounts of speaker-dependent data and retrain the recognizer models. Similarly, it is desirable to avoid the expense of collecting additional data when the recognizer is going to be used on speech transmitted through a different channel than the one used in training. Such problems may be solved by adapting the recognizer models, using much smaller amounts of adaptation data than those used in conventional training techniques. We have developed an adaptation technique that combines the advantages of the two main adaptation

approaches used in the past — namely, the quick adaptation characteristics of transformation approaches and the nice asymptotic properties of Bayesian schemes.

3.6.1 Transformation-based Adaptation

Transformation-based approaches to speaker adaptation transform the speaker's feature space to "match" the space of the training population. This approach has the advantage of simplicity and, if the number of free parameters is small, transformation techniques adapt to the user with only a small amount of adaptation speech (quick adaptation). Disadvantages of transformation methods are that they are usually text-dependent — that is, they require the new speaker to record some predetermined sentences — and that they may not take full advantage of large amounts of adaptation data.

We have developed a novel transformation-based approach to speaker adaptation for continuous mixture-density HMMs. We apply the transformation at the distribution level, instead of transforming the feature vectors directly, and we use the expectation-maximization (EM) algorithm to estimate the transformation parameters by maximizing the likelihood of the adaptation data. Using this approach, we are not required to time-align the new- and reference-speaker data, and the transformation parameters can be estimated using new-speaker data alone. Our scheme can also be viewed as a constrained estimation of Gaussian mixtures, since we apply the same transformation to all the components of a particular mixture (or a group of mixtures, if there is tying of transformations) instead of independently reestimating them. This approach achieves quick adaptation by adapting Gaussians for which there were no observations in the training data, based on data that were most likely generated by other Gaussians of the same or neighboring mixtures. A paper describing this technique is included in the appendix.

3.6.2 Combined Bayesian and Transformation Methods

A second main family of adaptation algorithms follows a Bayesian approach, where the speaker-independent information is encapsulated in the prior distributions. The Bayesian approach has nice asymptotic properties: speaker-adaptive performance will converge to speaker-dependent performance as the amount of adaptation speech increases. However, the adaptation rate is usually slow, since only the Gaussians of the speaker-independent models that are most likely to have generated some of the adaptation data will be adapted to the speaker. These Gaussians may represent only a small fraction of the total number in continuous HMMs with a large number of Gaussians.

We have developed an adaptation scheme that retains the nice properties of Bayesian schemes for large amounts of adaptation data, and has improved performance for small amounts of adaptation data. We have achieved this by using our transformation-based adaptation as a preprocessing step to transform the speaker-independent models so that they better match the new speaker characteristics and improve the prior information in Bayesian estimation schemes. A paper describing this technique is included in the appendix.

3.6.3 Hierarchical Transformations

The transformation-based algorithm described in Section 3.6.1 results in quick adaptation of the HMMs. To reach optimum performance for a given set of adaptation sentences we need to determine the total number of parameters used in the transformations. An excessive number of parameters will result in some of them being undertrained and therefore hurting performance. A conservatively low number of parameters will guarantee that the recognition error rate is bounded by the speaker-independent performance but will not use the adaptation data efficiently.

To alleviate this problem we invented a hierarchical adaptation scheme that will assign the most specific transformation to each genone. A global threshold is provided to guarantee that only transformations that get enough adaptation training frames are used to transform the genones. In this hierarchical scheme we estimate transformations following a tree structure, where at the top we have a global transformation (estimated with all the adaptation data) and at the leaves we have the most specific transformations (one for each genone). To find the optimum transformation for a given genone we search for the most specific transformation (close to the leaf) that is above the threshold. We found that the hierarchical adaptation scheme results in very robust systems. For example, we tested the algorithm using various adaptation sets ranging from 1 to 40 sentences. Without the hierarchical method the speaker-adapted performance may become worse than the speaker-independent case for very short amounts of adaptation data. This does not happen when the hierarchical method is used. This characteristic is very important for the global consistency techniques that we are currently investigating in the follow-on SRI project 6429.

3.6.4 Speaker Adaptation Results

We evaluated our adaptation algorithms on the Spoke 3 task of the Phase 1, large-vocabulary WSJ corpus, trying to improve recognition performance for nonnative speakers of American English. We measured the word error rate on the development set and the November 1993 ARPA evaluation set of the WSJ corpus using a trigram language model. Our results, presented in Table 7, represent the best reported results to date on this task. The nonnative recognition performance after adaptation, using only 40 sentences, is slightly higher than that of native speakers, which for the same speaker-independent models is a 9.7% and 7.2% word error with a bigram and a trigram language model, respectively.

Test Set	Speaker Independent		Speaker Adapted	
	Bigram	Trigram	Bigram	Trigram
Development	29.3	23.5	13.6	10.3
November 1993	21.0	16.5	13.0	10.0

Table 7. Adaptation results using bigram and trigram language models on various test sets of nonnative speakers

3.7 RECOGNITION IN ADDITIVE NOISE

A noise-robust recognizer was designed to operate in a noisy channel. We studied several approaches, including POF mapping, HMM adaptation, and combination of mapping and adaptation.

The experimental framework was defined according to the 1994 ARPA-sponsored CSR evaluation Spoke test 10 (noisy channel). In this test a 1-minute sample of the noise is provided for the adaptation of the recognizer models. This noise was used to build several stereo compensation data sets consisting of a clean channel and a noisy channel at a given SNR level.

The compensation sets were used to estimate POF mappings and to adapt the HMMs to a specific SNR level. The mapping technique estimates the clean features at the front-end signal processing stage, while the HMM adaptation method modifies the HMMs to match the noise level of the test sentence.

We found that both approaches are very effective in compensating for the additive noise when used alone. We also found that as the noise level increases (lower SNR) combining both techniques produces lower error rates. In the 1994 ARPA benchmarks we showed that the ratio of noisy-speech error rate over the clean-speech error rate can be reduced from 4.9 to 1.8 for low SNR levels using this compensation scheme. A paper describing this technique is included in the appendix.

3.8 GLOBAL CONSISTENCY MODELING

We have recently initiated research into global consistency modeling on ARPA-funded SRI project 6429. In Section 3.3 we represented the local history of a particular frame, and conditioned the likelihood of each frame on this local history. In global consistency modeling the history involves longer periods of time, such as previous sentences or even sentences in the training data. For example, if we had training data from different speakers, then the global history might be allowed to take on values of each different training speaker. In this case, the likelihood of a test utterance must be conditioned separately on each training speaker, and the model of the training speaker corresponding to the maximum likelihood is used. This idea is already used in many speech recognition systems where separate male and female models are trained, and during recognition the model that gives higher likelihood is used.

We have initiated research into the above idea of training a multitude of template models, and during testing, picking the best set of models to use for recognition. Among the research issues are model storage, efficient computation of the best models, and optimal estimation of the model for recognition. We have used ideas from the speaker adaptation algorithm described in Section 3.6 to address the storage issues. Thus, each template model is speaker-adapted instead of speaker-dependent. This greatly reduces the storage requirements. We have also developed some ideas for efficient computation of the best models among the templates and estimation of the appropriate recognition model. Based on these ideas, we were able to reduce the speaker-independent word error rate by about 5% on a test set of 230 WSJ sentences from the 1993

development and evaluation set. Details of this method are presented in the December 1994 quarterly report of SRI project 6429.

4. REFERENCES

Algazi93 Algazi, V. R., K. L. Brown, M. J. Ready, D. H. Irvine, C. L. Cadwell, and S. Chung, "Transform Representation of the Spectra of Acoustic Speech Segments with Applications - I: General Approach and Application to Speech Recognition," *IEEE Trans. Speech and Audio Processing*, Vol. 1(2), pp. 180-195, April 1993.

Bocchieri93 Bocchieri, E., "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," *Proc. ICASSP*, pp. II-692 - II-695, April 1993.

Brown87 Brown, P.F., *The Acoustic Modeling Problem in Automatic Speech Recognition*, Ph.D. Thesis, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, May 1, 1987, CMU-CS-87-125.

Digalakis93 Digalakis, V., J. R. Rohlicek, and M. Ostendorf, "ML Estimation of a Stochastic Linear System with the EM Algorithm and its Application to Speech Recognition," to appear in *IEEE Trans. Speech and Audio Processing*, October 1993.

Digalakis94 Digalakis, V., and H. Murveit, "High-Accuracy Large-Vocabulary Speech Recognition Using Mixture-Tying and Consistency Modeling," *ARPA Human Language Technology Workshop,* March 1994.

Kannan94 Kannan, A., M. Ostendorf, and J. R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," in *IEEE Trans. Speech and Audio Processing*, July 1994.

Katz87 Katz, S., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *ASSP-35* pp. 400-401, March 1987.

Kenny90 Kenny, P., M. Lennig, and P. Mermelstein, "A Linear Predictive HMM for Vector-Valued Observations with Applications to Speech Recognition," *IEEE Trans. Acoustics, Speech and Signal Processing,* Vol. ASSP-38(2), pp. 220-225, February 1990.

Murveit94 Murveit, H., P. Monaco, V. Digalakis, and J. Butzberger, "Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System," *ARPA Human Language Technology Workshop,* March 1994.

Ostendorf89 Ostendorf, M., and S. Roukos, "A Stochastic Segment Model for Phoneme-based Continuous Speech Recognition," *IEEE Trans. Acoustics, Speech and Signal Processing,* Vol. ASSP-37(12), pp. 1857-1869, December 1989.

Wellekens87 Wellekens, C., "Explicit Time Correlation in Hidden Markov Models for Speech Recognition," *Proc. ICASSP, 1987.*

5. PUBLICATIONS, PRESENTATIONS AND REPORTS

5.1 REFEREED PAPERS PUBLISHED

L. Neumeyer, V. Digalakis, and M. Weintraub, "Training Issues and Channel Equalization Techniques for the Construction of Telephone Acoustic Models Using a High-Quality Speech Corpus," in *IEEE Trans. Speech and Audio Processing,* Special Issue, October 1994.

5.2 REFEREED PAPERS SUBMITTED BUT NOT YET PUBLISHED

V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," submitted to *IEEE Trans. Speech and Audio Processing,* June 1994.

V. Digalakis, D. Rtischev, and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures", Submitted to *IEEE Trans. Speech and Audio Processing*, April 1994.

V. Digalakis and L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," Submitted to *IEEE Trans. Speech and Audio Processing*, December 1995.

5.3 PAPERS PUBLISHED IN REFEREED CONFERENCE PROCEEDINGS[9]

H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive-Search Techniques," *IEEE International Conference on Acoustics, Speech and Signal Processing,* April 1993.

H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Progressive Search for Large Vocabulary Speech Recognition," *ARPA Human Language Technology Workshop*, March 1993.

V. Digalakis, P. Monaco, and H. Murveit, "Acoustic Calibration and Search in SRI's Large Vocabulary Speech Recognition System," *IEEE ASR Workshop*, December 1993.

V. Digalakis and H. Murveit, "High-Accuracy Large-Vocabulary Speech Recognition Using Mixture-Tying and Consistency Modeling," *ARPA Human Language Technology Workshop,* March 1994.

H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger, "Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System," *ARPA Human Language Technology Workshop,* March 1994.

---

9. These papers had extended abstracts that were refereed; the papers themselves were not refereed.

L. Neumeyer and M. Weintraub, "Microphone-Independent Robust Signal Processing Using Probabilistic Optimum Filtering", *ARPA Human Language Technology Workshop*, March 1994.

M. Weintraub, L. Neumeyer, and V. Digalakis, "SRI November 1993 CSR Spoke Evaluation," *ARPA Spoken Language Systems Technology Workshop*, March 1994.

V. Digalakis and H. Murveit, "An Algorithm for Optimizing the Degree of Mixture-Tying in a Large-Vocabulary HMM-Based Speech Recognizer," *IEEE International Conference on Acoustics, Speech and Signal Processing,* April 1994.

L. Neumeyer and M. Weintraub, "Probabilistic Optimum Filtering for Robust Speech Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 1994.

V. Digalakis, D. Rtischev, and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," *Speech Research Symposium XIV,* June 1994.

L. Neumeyer and M. Weintraub, "Robust Speech Recognition in Noise Using Mapping and Adaptation Techniques," to appear in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1995.

V. Digalakis and L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," to appear in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1995.

5.4 INVITED PRESENTATIONS:

H. Murveit, "Progressive Search Techniques," presented at the ARPA Spoken Language Systems Technology Workshop, January 1993, Massachusetts Institute of Technology, Cambridge, MA.

M. Weintraub, "SRI's Stress-Test Benchmark," presented at the ARPA Spoken Language Systems Technology Workshop, January 1993, Massachusetts Institute of Technology, Cambridge, MA.

Demonstration of a 20,000-word continuous speech recognition in ARPA's Wall Street Journal domain, ARPA Spoken Language Systems Technology Workshop, January 1993, Massachusetts Institute of Technology, Cambridge, MA.

M. Cohen, V. Digalakis, H. Murveit, P. Price, and M. Weintraub, "Speech Recognition: an Overview, Examples and Demonstration," presented at Information Systems Laboratory, Stanford University, February 1993.

H. Murveit organized and gave overview and summary talk for the demonstration session of the ARPA Human Language Technology Workshop, March 22, 1993, Plainsboro, NJ.

M. Weintraub, "Progressive Search for Large Vocabulary Speech Recognition," presented at the ARPA Human Language Technology Workshop, March 22, 1993, Plainsboro, New Jersey.

H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER™ Speech Recognition System: Progressive-Search Techniques," presented at ICASSP-93, April 1993.

V. Digalakis, "Search and Modeling Issues in Large-Vocabulary Speech Recognition" presented at Xerox PARC, August 1993.

V. Digalakis, P. Monaco, and H. Murveit, "Acoustic Calibration and Search in SRI's Large Vocabulary Speech Recognition System," presented at the IEEE ASR workshop, Snowbird, UT, December 1993.

V. Digalakis and H. Murveit, "High-Accuracy Large-Vocabulary Speech Recognition at SRI International," presented at the ARPA Human Language Technology Workshop, March 1994, Plainsboro, NJ.

V. Digalakis, H. Murveit, P. Monaco, H. Bratt, J. Butzberger and M. Weintraub, "SRI November 1993 CSR Hub Evaluation," presented at the ARPA SLT Workshop, March 1994, Plainsboro, NJ.

V. Digalakis, D. Rtischev, and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," presented at the Speech Research Symposium XIV, June 1994.

V. Digalakis chaired the Large-Vocabulary Speech Recognition II Section at the IEEE International Conference on Acoustics, Speech and Signal Processing, April 1994.

V. Digalakis and H. Murveit, "An Algorithm for Optimizing the Degree of Mixture-Tying in a Large-Vocabulary HMM-Based Speech Recognizer," presented at the IEEE International Conference on Acoustics, Speech and Signal Processing, April 1994.

6. TRAVEL

This project has partially of fully supported the participation of SRI staff members in many important technical and professional meetings:

- The DARPA-sponsored HLT workshop in Princeton, NJ, in March 1993 was attended by Vassilios Digalakis, Hy Murveit, Mitch Weintraub, and Victor Abrash.
- The ARPA-sponsored HLT/SLT workshop in Princeton, NJ, in March 1994 was attended by Vassilios Digalakis, Hy Murveit, Mitch Weintraub, Patti Price, and Peter Monaco.
- Vassilios Digalakis, Hy Murveit, and Mitch Weintraub attended the Spoken Language System Technology workshop in Cambridge, MA, in January 1993.
- Hy Murveit attended the colloquium on Human-Machine Communication by Voice in Los Angeles, CA, in February 1993.
- Vassilios Digalakis, Hy Murveit, and Mitch Weintraub attended the IEEE International Conference on Acoustics, Speech, and Signal Processing, in Minneapolis, MN, in April 1993.
- The ARPA planning meeting in Washington, DC, in June 1993, was attended by Patti Price.
- The ARPA meeting in Pittsburgh, PA, in September 1993, was attended by Patti Price.
- The Workshop on Robust Speech Analysis, at Rutgers University, NJ, was attended by Vassilios Digalakis in July 1993.
- Vassilios Digalakis, Peter Monaco, and Hy Murveit attended the December 1993 IEEE workshop on Automatic Speech Recognition at Snowbird, UT.
- The IEEE International Conference on Acoustics, Speech, and Signal Processing, in Adelaide, Australia, in April 1994, was attended by Vassilios Digalakis, and Peter Monaco.
- Patti Price attended the ARPA coordinating committee meeting in Washington, DC, in June 1994.
- Vassilios Digalakis attended the 2nd Workshop on Robust Speech Analysis at Rutgers University, NJ, in August 1994.

7. TRANSITIONS AND DOD INTERACTIONS

We were active participants in the two 6-week Robust Speech Processing workshops sponsored by NSA at Rutgers in July-August 1993 and June-August 1994. During the first year, two researchers, Leo Neumeyer and Vassilios Digalakis, focused on training issues and channel equalization techniques for acoustic modeling of telephone speech, and their work at the workshop was included in a special issue of the *IEEE Transactions on Speech and Audio Processing* published in October 1994. During the second year, they focused on techniques for unsupervised speaker adaptation.

SRI's DECIPHER speech recognition technology has being transitioned to Boston University for joint research funded by NSF and ARPA. Under internal SRI funding our DECIPHER technology was modified to support ARPA-sponsored research on robust front-end signal processing at CAIP in collaboration with laboratories at the David Sarnoff Research Center.

Several applications based on DECIPHER technology were demonstrated at Spoken Language Technology Applications Day in April 1993. This event was attended by more than 300 people, about equally divided among government and commercial representatives. Our participation in this event was sponsored by internal funds.

SRI has invested significant internal resources toward the development of robust, portable speech recognition software and tools for its use and has launched a spin-off company, Corona Corporation, to commercialize speech recognition technology developed at SRI's Speech Technology and Research (STAR) Laboratory. Two of this project's main contributors, Hy Murveit and Peter Monaco, are among the founders of Corona which is currently owned by SRI, Corona employees and STAR Lab employees. SRI and the STAR Lab maintain a close relationship with Corona. The STAR Lab's main focus is on research, development and advancing the technology. Corona's main focus is on applications, product development and exploitation of the technology commercially. Several commercial clients are using the resultant technology in their own research or in field trials.

8. SOFTWARE AND HARDWARE PROTOTYPES

The algorithms and software developed in this project have been incorporated into the DECIPHER speech recognition system. We are attempting to commercialize speech recognition based on DECIPHER technology and based on tools and other extensions to it that were funded by SRI's support. SRI currently has several commercial clients that are in the process of evaluating speech recognition products based on DECIPHER technology.

9. APPENDIX

PUBLICATIONS COVERING PROJECT RESEARCH

H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER™ Speech Recognition System: Progressive-Search Techniques," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993.*

V. Digalakis, P. Monaco, and H. Murveit, "Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers," submitted to *IEEE Trans. Speech and Audio Processing,* June 1994.

V. Digalakis and H. Murveit, "High-Accuracy Large-Vocabulary Speech Recognition Using Mixture-Tying and Consistency Modeling," *ARPA Human Language Technology Workshop,* March 1994.

H. Murveit, P. Monaco, V. Digalakis, and J. Butzberger, "Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System," *ARPA Human Language Technology Workshop,* March 1994.

M. Weintraub, L. Neumeyer, and V. Digalakis, "SRI November 1993 CSR Spoke Evaluation," *ARPA Spoken Language Systems Technology Workshop*, March 1994.

V. Digalakis, D. Rtischev, and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," Submitted to *IEEE Trans. Speech and Audio Processing*, April 1994.

V. Digalakis and L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," Submitted to *IEEE Trans. Speech and Audio Processing*, Dec. 1995.

L. Neumeyer, V. Digalakis, and M. Weintraub, "Training Issues and Channel Equalization Techniques for the Construction of Telephone Acoustic Models Using a High-Quality Speech Corpus," *IEEE Trans. Speech and Audio Processing,* Special Issue, October 1994.

V. Digalakis and L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," to appear in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1995.

L. Neumeyer and M. Weintraub, "Robust Speech Recognition in Noise using Mapping and Adaptation Techniques," to appear in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 1995.

LARGE-VOCABULARY DICTATION USING SRI'S DECIPHER™ SPEECH RECOGNITION SYSTEM: PROGRESSIVE SEARCH TECHNIQUES

Hy Murveit
John Butzberger
Vassilios Digalakis
Mitch Weintraub

SRI International

ABSTRACT

We describe a technique we call *Progressive Search* which is useful for developing and implementing speech recognition systems with high computational requirements. The scheme iteratively uses more and more complex recognition schemes, where each iteration constrains the search space of the next. An algorithm, the *Forward-Backward Word-Life Algorithm*, is described. It can generate a word lattice in a progressive search that would be used as a language model embedded in a succeeding recognition pass to reduce computation requirements. We show that speed-ups of more than an order of magnitude are achievable with only minor costs in accuracy.

1. INTRODUCTION

Many advanced speech recognition techniques cannot be developed or used in practical speech recognition systems because of their extreme computational requirements. Simpler speech recognition techniques can be used to recognize speech in reasonable time, but they compromise word recognition accuracy. In this paper we aim to improve the speed/accuracy trade-off in speech recognition systems using progressive search techniques.

We define *progressive search* techniques as those which can be used to efficiently implement other, computationally burdensome techniques. They use results of a simple and fast speech recognition technique to constrain the search space of a following more accurate but slower running technique. This may be done iteratively—each progressive search pass uses a previous pass' constraints to run more efficiently, and provides more constraints for subsequent passes.

We will refer to the faster speech recognition techniques as "earlier-pass techniques", and the slower more accurate techniques as "advanced techniques." Constraining the costly advanced techniques in this way can make them run significantly faster without significant loss in accuracy.

The key notions in progressive search techniques are:

1. An early-pass speech recognition phase builds a lattice, which contains all the likely recognition unit strings (e.g. word sequences) given the techniques used in that recognition pass.
2. A subsequent pass uses this lattice as a grammar that constrains the search space of an advanced technique (e.g., only the word sequences contained in a word lattice of pass p would be considered in pass p+1).

Allowing a sufficient breadth of lattice entries should allow later passes to recover the correct word sequence, while ruling out very unlikely sequences, thus achieving high accuracy and high speed speech recognition.

2. PRIOR ART

There are three important categories of techniques that aim to solve problems similar to the ones the progressive search techniques target.

2.1. Fast-Match Techniques

Fast-match techniques[1] are similar to progressive search in that a coarse match is used to constrain a more advanced computationally burdensome algorithm. The fast match, however, simply uses the local speech signal to constrain the costly advanced technique. Since the advanced techniques may take advantage of non-local data, the accuracy of a fast-match is limited and will ultimately limit the overall technique's performance. Techniques such as progressive search can bring more global knowledge to bear when generating constraints, and, thus, more effectively speed up the costly techniques while retaining more of their accuracy.

2.2. N-Best Recognition Techniques

N-best techniques[2] are also similar to progressive search in that a coarse match is used to constrain a more computationally costly technique. In this case, the coarse matcher is a complete (simple) speech recognition system. The output of the N-best system is a list of the top N most likely sentence hypotheses, which can then be evaluated with the slower but more accurate techniques.

Progressive search is a generalization of N-best—the earlier-pass technique produces a graph, instead of a list of N-best sentences. This generalization is crucial because N-best is only computationally effective for N in the order of tens or hundreds. A progressive search word graph can effectively account for orders of magnitude more sentence hypotheses. By limiting the advanced techniques to just searching the few top N sentences, N-best is destined to limit the effectiveness of the advanced techniques and, consequently, the overall system's

accuracy. Furthermore, it does not make much sense to use N-best in an iterative fashion as it does with progressive searches.

2.3. Word Lattices

This technique is the most similar to progressive search. In both approaches, an initial-pass recognition system can generate a lattice of word hypotheses. Subsequent passes can search through the lattice to find the best recognition hypothesis. It should be noted that, although we refer to lattices as word lattices, they could be used at other linguistic level, such as the phoneme, syllable, e.t.c.

In the traditional word-lattice approach, the word lattice is viewed as a scored graph of possible segmentations of the input speech. The lattice contains information such as the acoustic match between the input speech and the lattice word, as well as segmentation information.

The progressive search lattice is not viewed as a scored graph of possible segmentations of the input speech. Rather, the lattice is simply viewed as a word-transition grammar which constrains subsequent recognition passes. Temporal and scoring information is intentionally left out of the progressive search lattice.

This is a critical difference. In the traditional word-lattice approach, many segmentations of the input speech which could not be generated (or scored well) by the earlier-pass algorithms will be eliminated for consideration before the advanced algorithms are used. With progressive-search techniques, these segmentations are implicit in the grammar and can be recovered by the advanced techniques in subsequent recognition passes.

3. Building Progressive Search Lattices

The basic step of a progressive search system is using a speech recognition algorithm to make a lattice which will be used as a grammar for a more advanced speech recognition algorithm. This section discusses how these lattices may be generated. We focus on generating word lattices, though these same algorithms are easily extended to other levels.

3.1. The Word-Life Algorithm

We implemented the following algorithm to generate a word-lattice as a by-product of the beam search used in recognizing a sentence with the DECIPHER™ system[4-7].

1. For each frame, insert into the table *Active(W, t)* all words W active for each time t. Similarly construct tables *End(W, t)* and *Transitions(W_1, W_2, t)* for all words ending at time t, and for all word-to-word transition at time t.

2. Create a table containing the word-lives used in the sentence, *WordLives(W, T_{start}, T_{end})*. A *word-life* for word W is defined as a maximum-length interval (frame T_{start} to T_{end}) during which some phone in word W is active. That is,
$$W \in Active(W, t), T_{start} \le t \le T_{end}$$

3. Remove word-lives from the table if the word never ended between T_{start} and T_{end}, that is, remove *WordLives(W, T_{start}, T_{end})* if there is time t between T_{start} and T_{end} where *End(W, t)* is true.

4. Create a finite-state graph whose nodes correspond to word-lives, whose arcs correspond to word-life transitions stored in the *Transitions* table. This finite state graph, augmented by language model probabilities, can be used as a grammar for a subsequent recognition pass in the progressive search.

This algorithm can be efficiently implemented, even for large vocabulary recognition systems. That is, the extra work required to build the "word-life lattice" is minimal compared to the work required to recognize the large vocabulary with a early-pass speech recognition algorithm.

This algorithm develops a grammar which contains all whole-word hypotheses the early-pass speech recognition algorithm considered. If a word hypothesis was active and the word was processed by the recognition system until the word finished (was not pruned before transitioning to another word), then this word will be generated as a lattice node. Therefore, the size of the lattice is directly controlled by the recognition search's beam width.

This algorithm, unfortunately, does not scale down well—it has the property that small lattices may not contain the best recognition hypotheses. This is because one must use small beam widths to generate small lattices. However, a small beam width will likely generate pruning errors.

Because of this deficiency, we have developed the Forward/Backward Word-Life Algorithm described below.

3.2. Extending the Word-Life Algorithm Using Forward And Backward Recognition Passes

We wish to generate word lattices that scale down gracefully. That is, they should have the property that when a lattice is reduced in size, the most likely hypotheses remain and the less likely ones are removed. As was discussed, this is not the case if lattices are scaled down by reducing the beam search width.

The forward-backward word-life algorithm achieves this scaling property. In this new scheme, described below, the size of the lattice is controlled by the *LatticeThresh* parameter.

1. A standard beam search recognition pass is done using the early-pass speech recognition algorithm. (None of the lattice building steps from Section 3.1 are taken in this forward pass).

2. During this forward pass, whenever a transition leaving word W is within the beam-search, we record that probability in *ForwardProbability(W frame)*.

3. We store the probability of the best scoring hypothesis from the forward pass, *Pbest*, and compute a pruning value
Pprune = Pbest / LatticeThresh.

4. We then recognize the same sentence over again using the same models, but the recognition algorithm is run backwards[1].

5. The lattice building algorithm described in Section 3.1 is used in this backward pass with the following exception. During the backward pass, whenever there is a transition between words W_i and W_j at time t, we compute the overall hypothesis probability P_{hyp} as the product of *ForwardProbability*(W_j,t-1), the language model probability $P(W_i|W_j)$, and the Backward pass probability that W_i ended at time t (i.e. the probability of starting *word* W_i at time t and finishing the sentence). If $P_{hyp} < P_{prune}$, then the backward transition between W_i and W_j at time t is blocked.

Step 5 above implements a backwards pass pruning algorithm. This both greatly reduces the time required by the backwards pass, and adjusts the size of the resultant lattice.

4. Progressive Search Lattices

We have experimented with generating word lattices where the early-pass recognition technique is a simple version of the DECIPHER™ speech recognition system, a 4-feature, discrete density HMM trained to recognize a 5,000 vocabulary taken from DARPA's WSJ speech corpus. The test set is a difficult 20-sentence subset of one of the development sets.

We define the number of errors in a single path p in a lattice, *Errors(p)*, to be the number of insertions, deletions, and substitutions found when comparing the words in p to a reference string. We define the number of errors in a word lattice to be the minimum of *Errors(p)* for all paths p in the word lattice.

The following tables show the effect adjusting the beam width and *LatticeThresh* has on the lattice error rate and on the lattice size (the number of nodes and arcs in the word lattice). The grammar used by the has approximately 10,000 nodes and 1,000,000 arcs. The the simple recognition system had a 1-best word error-rate ranging from 27% (beam width 1e-52) to 30% (beam width 1e-30).

Table 1: Effect Of Pruning On Lattice Size

Beam Width 1e-30

Lattice Thresh	nodes	arcs	# errors	%word error
1e-5	60	278	43	10.57
1e-9	94	541	34	8.35
1e-14	105	1016	30	7.37
1e-18	196	1770	29	7.13
1e-32	323	5480	23	5.65
1e-45	372	8626	23	5.65
inf	380	9283	23	5.65

Beam Width 1e-34

Lattice Thresh	nodes	arcs	# errors	%word error
1e-5	64	299	28	6.88
1e-9	105	613	20	4.91
1e-14	141	1219	16	3.93
1e-18	260	2335	15	3.69
1e-23	354	3993	15	3.69
1e-32	537	9540	15	3.69

Beam Width 1e-38

Lattice Thresh	nodes	arcs	# errors	%word error
1e-14	186	1338	14	3.44
1e-18	301	2674	13	3.19
1e-23	444	4903	12	2.95

Beam Width 1e-42

Lattice Thresh	nodes	arcs	# errors	%wd error
1e-14	197	1407	13	3.19
1e-18	335	2926	11	2.70
1e-23	520	5582	10	2.46

Beam Width 1e-46

Lattice Thresh	nodes	arcs	# errors	%word error
1e-14	201	1436	13	3.19
1e-18	351	3045	10	2.46
1e-23	562	5946	10	2.46

Beam Width 1e-52

Lattice Thresh	nodes	arcs	# errors	%word error
1e-14	216	1582	12	2.95
1e-18	381	3368	9	2.21

The two order of magnitude reduction in lattice size has a significant impact on HMM decoding time. Table 2 shows the per-sentence computation time required for the above test set when computed using a Sparc2 computer, for both the original grammar, and word lattice grammars generated using a *LatticeThresh* of 1e-23.

1. Using backwards recognition the sentence is processed from last frame to first frame with all transitions reversed.

Table 2: Lattice Computation Reductions

Beam Width	Forward pass recognition time (secs)	Lattice recognition time (secs)
1e-30	167	10
1e-34	281	16
1e-38	450	24
1e-46	906	57
1e-52	1749	65

5. Applications of Progressive Search Schemes

Progressive search schemes can be used in the same way N-best schemes are currently used. The two primary applications we've had at SRI are:

5.1. Reducing the time required to perform speech recognition experiments

At SRI, we've been experimenting with large-vocabulary tied-mixture speech recognition systems. Using a standard decoding approach, and average decoding times for recognizing speech with a 5,000-word bigram language model were 46 times real time. Using lattices generated with beam widths of 1e-38 and a *LatticeThresh* of 1e-18 we were able to decode in 5.6 times real time). Further, there was no difference in recognition accuracy between the original and the lattice-based system.

5.2. Implementing recognition schemes that cannot be implemented with a standard approach.

We have implemented a trigram language model on our 5,000-word recognition system. This would not be feasible using standard decoding techniques. Typically, continuous-speech trigram language models are implemented either with fastmatch technology or, more recently, with N-best schemes. However, it has been observed at BBN that using an N-best scheme (N=100) to implement a trigram language model for a 20,000 word continuous speech recognition system may have significantly reduced the potential gain from the language model. That is, about half of the time, correct hypotheses that would have had better (trigram) recognition scores than the other top-100 sentences were not included in the top 100 sentences generated by a bigram-based recognition system[8].

We have implemented trigram-based language models using word-lattices, expanding the finite-state network as appropriate to unambiguously represent contexts for all trigrams. We observed that the number of lattice nodes increased by a factor of 2-3 and the number of lattice arcs increased by a factor of approximately 4 (using lattices generated with beam widths of 1e-38 and a *LatticeThresh* of 1e-18). The resulting decoding times increased approximately by 50% when using trigram lattices instead of bigram lattices.

ACKNOWLEDGEMENTS

We gratefully acknowledge support for this work from DARPA through Office of Naval Research Contract N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the government funding agencies.

REFERENCES

1. Bahl, L.R., de Souza, P.V., Gopalakrishnan, P.S., Nahamoo, D., and M. Picheny, "A Fast Match for Continuous Speech Recognition Using Allophonic Models," *1992 IEEE ICASSP*, pp. I-17-21.
2. Schwartz, R., Austin, S., Kubala, F., Makhoul, J., Nguyen, L., Placeway, P., and G. Zavaliagkos, "New uses for the N-Best Sentence Hypotheses Within the BYBLOS Speech Recognition System", *1992 IEEE ICASSP*, pp. I-1-4.
3. Chow, Y.L., and S. Roukos, "Speech Understanding Using a Unification Grammar", *1989 IEEE ICASSP*, pp. 727-730
4. H. Murveit, J. Butzberger, and M. Weintraub, "Performance of SRI's DECIPHER Speech Recognition System on DARPA's CSR Task," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp 410-414
5. Murveit, H., J. Butzberger, and M. Weintraub, "Reduced Channel Dependence for Speech Recognition," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp. 280-284.
6. H. Murveit, J. Butzberger, and M. Weintraub, "Speech Recognition in SRI's Resource Management and ATIS Systems," 1991 DARPA Speech and Natural Language Workshop, pp. 94-100.
7. Cohen, M., H. Murveit, J. Bernstein, P. Price, and M. Weintraub, "The DECIPHER™ Speech Recognition System," *1990 IEEE ICASSP*, pp. 77-80.
8. Schwartz, R., BBN Systems and Technologies, Cambridge MA, Personal Communication

Genones: Generalized Mixture Tying in Continuous Hidden Markov Model-Based Speech Recognizers[1]

V. Digalakis 415-859-5540 vas@speech.sri.com

P. Monaco 415-859-4927 monaco@speech.sri.com

H. Murveit 415-859-5447 hy@speech.sri.com

SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
Fax: 415-859-5984

May 20, 1994

EDICS SA 1.6.10 and SA 1.6.11

ABSTRACT

An algorithm is proposed that achieves a good trade-off between modeling resolution and robustness by using a new, general scheme for tying of mixture components in continuous mixture-density hidden Markov model (HMM)-based speech recognizers. The sets of HMM states that share the same mixture components are determined automatically using agglomerative clustering techniques. Experimental results on ARPA's Wall-Street Journal corpus show that this scheme reduces errors by 25% over typical tied-mixture systems. New fast algorithms for computing Gaussian likelihoods—the most time-consuming aspect of continuous-density HMM systems—are also presented. These new algorithms significantly reduce the number of Gaussian densities that are evaluated with little or no impact on speech recognition accuracy.

---

1. This research was supported by the Advanced Research Projects Agency under Contract ONR N00014-92-C-0154. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Project Agency or the National Science Foundation.

1 INTRODUCTION

Hidden Markov model (HMM)-based speech recognizers with *tied-mixture* (TM) observation densities [1][2] achieve robust estimation and efficient computation of the density likelihoods. However, the typical mixture size used in TM systems is small and does not accurately represent the acoustic space. Increasing the number of the mixture components (also known as the codebook size) is not a feasible solution, since the mixture-weight distributions become too sparse. In large-vocabulary problems, where a large number of basic HMMs is used and each has only a few observations in the training data, sparse mixture-weight distributions cannot be estimated robustly and are expensive to store.

HMMs with continuous mixture densities and no tying constraints (*fully continuous* HMMs), in contrast, provide a detailed stochastic representation of the acoustic space at the expense of increased computational complexity and lack of robustness. A detailed representation is critical for large-vocabulary speech recognition. It has recently been shown [3] that, in large-vocabulary recognition tasks, HMMs with continuous mixture densities and no tying consistently outperform HMMs with tied-mixture densities. To overcome the robustness issue, continuous HMM systems use various schemes. Gauvain in [4] smooths the mixture-component parameters with maximum *a-posteriori* (MAP) estimation and implicitly clusters models that have small amounts of training via back-off mechanisms. Young in [5] uses clustering at the HMM state level and estimates mixture densities only for clustered states with enough observations in the training data.

In this work, and in order to achieve the optimum trade-off between acoustic resolution and robustness, we choose to generalize the tying of mixture components. From the fully continuous HMM perspective, we improve the robustness by sharing the same mixture components among arbitrarily defined sets of HMM states. From the tied-mixture HMM perspective, we improve the acoustic resolution by simultaneously increasing the number of different sets of mixture components (or codebooks) and reducing each codebook's size. These two changes can be balanced so that the total number of component densities in the system is effectively increased. We propose a new algorithm that automatically

determines the sets of HMM states that will share the same mixture components. The algorithm can also be viewed as a method that transforms a system with a high degree of tying among the mixture components to a system with a smaller degree of tying. The appropriate degree of tying for a particular task depends on the difficulty of the task, the amount of available training data, and the available computational resources for recognition, since systems with a smaller degree of tying have higher computational demands during recognition.

In Section 2 of this paper, we present the general form of mixture observation distributions used in HMMs and discuss previous work and variations of this form that have appeared in the literature. In Section 3 we present the main algorithm. In Section 4 we present word recognition results using ARPA's Wall Street Journal speech corpus. To deal with the increased amount of computation that continuous-density HMMs require during decoding, we present algorithms for the fast evaluation of Gaussian likelihoods in Section 5. Conclusions are given in Section 6.

2 MIXTURE OBSERVATION DENSITIES IN HMMs

A typical mixture observation distribution in an HMM-based speech recognizer has the form

$$p(x_t \mid s) = \sum_{q \in Q(s)} p(q|s) f(x_t|q) \quad . \tag{1}$$

where s represents the HMM state, x_t the observed feature at frame t, and $Q(s)$ the set of mixture-component densities used in state s. We shall use the term *codebook* to denote the set $Q(s)$. The stream of continuous vector observations can be modeled directly using Gaussians or other types of densities in the place of $f(x_t \mid q)$, and HMMs with this form of observation distributions appear in the literature as continuous HMMs [6].

Various forms of tying have appeared in the literature. When tying is not used, the sets of component densities are disjoint for different HMM states—that is, $Q(s) \cap Q(s') = \emptyset$

if $s \neq s'$. We shall refer to HMMs that use no sharing of mixture components as *fully continuous* HMMs.

To overcome the robustness and computation issues, the other extreme has also appeared in the literature: all HMM states share the same set of mixture components—that is, $Q(s) = Q$ is independent of the state s. HMMs with this degree of sharing were proposed in [1], [2] under the names *Semi-Continuous* and *Tied-Mixture* HMMs. Tied-mixture distributions have also been used with segment-based models, and a good review is given in [7]. The relative performance of tied-mixture and fully continuous HMMs usually depends on the amount of the available training data. With small to moderate amounts of training data, tied-mixture HMMs can be shown to outperform fully continuous ones, but with larger amounts of training data and appropriate smoothing fully continuous HMMs perform better [1][3].

Intermediate degrees of tying have also been examined. In phone-based tying, described in [8][9][10], only HMM states that belong to allophones of the same phone share the same mixture components—that is, $Q(s) = Q(s')$ if s and s' are states of context-dependent HMMs with the same center phone. We will use the term *phonetically tied* to describe this kind of tying. Of course, for context-independent models, phonetically tied and fully continuous HMMs are equivalent. However, phonetically tied mixtures (PTM) did not significantly improve recognition performance in previous work.

3 GENONIC MIXTURES

The continuum between fully continuous and tied-mixture HMMs can be sampled at any point. The choice of phonetically tied mixtures, although linguistically motivated, is somewhat arbitrary and may not achieve the optimum trade-off between resolution and trainability. We prefer to optimize performance by using an automatic procedure to identify subsets of HMM states that will share mixtures. The algorithm that we propose follows a bootstrap approach from a system that has a higher degree of tying (i.e., a TM or a PTM system), and progressively unties the mixtures using three steps: clustering, splitting and reestimation (Figure 1).

3.1 Clustering

In the first step of the algorithm (see Figure 1a), the HMM states of all allophones of a phone are clustered following an agglomerative hierarchical clustering procedure [11]. The states are clustered based on the similarity of their mixture-weight distributions. Any measure of dissimilarity between two discrete probability distributions can be used as the distortion measure during clustering. Following Lee [12] and Hwang [13], we use the increase in the weighted-by-counts entropy of the mixture-weight distributions that is caused by the merging of the two states. Let $H(s)$ denote the entropy of the discrete distribution $[p(q|s), q \in Q(s)]$,

$$H(s) = -\sum_{q \in Q(s)} p(q|s) \log p(q|s) \quad . \tag{2}$$

Then, the distortion that occurs when two states s_1 and s_2 with $Q(s_1) = Q(s_2)$ are clustered together into the clustered state s is defined as

$$d(s_1, s_2) = (n_1 + n_2) H(s) - n_1 H(s_1) - n_2 H(s_2) \quad , \tag{3}$$

where n_1, n_2 represent the number of observations used to estimate the mixture-weight distributions of the states s_1, s_2, respectively. The mixture-weight distribution of the clustered state s is

$$p(q|s) = \frac{n_1}{n_1 + n_2} p(q|s_1) + \frac{n_2}{n_1 + n_2} p(q|s_2) \, . \tag{4}$$

and the clustered state uses the same set of mixture components as the original states, $Q(s) = Q(s_1) = Q(s_2)$. This distortion measure can be easily shown to be nonnegative, and, in addition, $d(s, s) = 0$.

The clustering procedure partitions the set of HMM states S into disjoint sets of states

$$S = S_1 \cup S_2 \cup \ldots \cup S_n \quad . \tag{5}$$

where n, the number of clusters, is determined empirically.

The same codebook will be used for all HMM states belonging to a particular cluster S_i. Each state in the cluster will, however, retain its own set of mixture weights.

3.2 Splitting

Once the sets of HMM states that will share the same codebook are determined, seed codebooks for each set of states that will be used by the next reestimation phase are constructed (see Figure 1b). These seed codebooks can have a smaller number of component densities, since they are shared by fewer HMM states than the original codebook. They can be constructed by either one or a combination of two procedures:

- Identifying the most likely subset $Q(S_i) \subset Q(S)$ of mixture components for each cluster of HMM states S_i, and using a copy of that subset in the next phase as the seed codebook for states in S_i.
- Using a copy of the original codebook for each cluster of states. The number of component densities in each codebook can then be clustered down (see Section 5.1) after performing one iteration of the Baum-Welch algorithm over the training data with the new relaxed tying scheme.

The clustering and splitting steps of the algorithm define a mapping from HMM state to cluster index

$$g = \gamma(s) \quad , \tag{6}$$

as well as the set of mixture components that will be used by each state, $Q(s) = Q(g)$.

3.3 Reestimation

The parameters are reestimated using the Baum-Welch algorithm. This step allows the codebooks to deviate from the initial values (see Figure 1c) and achieve a better approximation of the distributions.

We shall refer to the Gaussian codebooks as *genones*[2] and to the HMMs with arbitrary tying of Gaussian mixtures as *genonic* HMMs. Clustering of either phone or subphone units in HMMs has also been used in [12][13][14][15]. Mixture-weight clustering of dif-

ferent HMM states can reduce the number of free parameters in the system and, potentially, improve recognition performance because of the more robust estimation. It cannot, however, improve the resolution with which the acoustic space is represented, since the total number of component densities in the system remains the same. In our approach, we use clustering to identify sets of subphonetic regions that will share mixture components. The subsequent steps of the algorithm increase the number of distinct densities in the system and provide the desired detail in the resolution.

Reestimation of the parameters can be achieved using the standard Baum-Welch reestimation formulae (see, e.g., [2] for the case of tied-mixture HMMs). For arbitrary tying of mixture components and Gaussian component densities, the observation distributions become

$$p(x_t \mid s) = \sum_{q \in Q(g)} p(q|s)\, N_{gq}(x_t ; \mu_{gq}, \Sigma_{gq}) \ , \tag{7}$$

where $N_{gq}(x_t ; \mu_{gq}, \Sigma_{gq})$ is the q-th Gaussian of genone g. It can be easily verified that the Baum-Welch reestimation formulae for the means and the covariances become

$$\hat{\mu}_{gq} = \frac{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \eta_t(j,q)\, x_t}{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \eta_t(j,q)} \tag{8}$$

and

$$\hat{\Sigma}_{gq} = \frac{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \eta_t(j,q)\ (x_t - \hat{\mu}_{gq})\ (x_t - \hat{\mu}_{gq})^T}{\sum_{s_j \in \gamma^{-1}(g)} \sum_t \eta_t(j,q)} \ , \tag{9}$$

where the first summation is over all states s_j in the inverse image $\gamma^{-1}(g)$ of the genonic index g. The accumulation weights in the equations above are

2. This term should be partially attributed to IBM's fenones and CMU's senones. A genone is a set of Gaussians shared by a set of states and should not be confused with the word genome.

$$\eta_t(j,q) = \left[\frac{\alpha_t(j)\,\beta_t(j)}{\sum_j \alpha_t(j)\,\beta_t(j)}\right]\left[\frac{p(q|s_j)\,N_{gq}(x_t;\mu_{gq},\Sigma_{gq})}{\sum_q p(q|s_j)\,N_{gq}(x_t;\mu_{gq},\Sigma_{gq})}\right], \quad (10)$$

where μ_{gq}, Σ_{gq} are the initial mean and covariance, the summations in the denominator are over all HMM states and all mixture components in a particular genone, respectively, and the quantities $\alpha_t(j)$, $\beta_t(j)$ are obtained using the familiar forward and backward recursions of the Baum-Welch algorithm [16]. The reestimation formulae for the remaining HMM parameters—i.e. mixture weights, transition probabilities, and initial probabilities—are the same as those presented in [2].

To reduce the large amount of computation involved in evaluating Gaussian likelihoods during recognition, we have developed fast computation schemes that are described in Section 5.

4 WORD RECOGNITION EXPERIMENTS

We evaluated genonic HMMs on the Wall Street Journal (WSJ) corpus [17]. We used SRI's DECIPHER ™ continuous speech recognition system, configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from an FFT filterbank. Context-dependent phonetic models were used, and the inventory of the triphones was determined by the number of occurrences of the triphones in the training data. In all of our experiments we used Gaussian distributions with diagonal covariance matrices as the mixture component densities. For fast experimentation, we used the progressive-search framework [18]. With this approach, an initial fast recognition pass creates word lattices for all sentences in the development set. These word lattices are used to constrain the search space in all subsequent experiments. In our development we used both the WSJ0 5,000-word and the WSJ1 64,000-word portions of the database. We used the baseline bigram and trigram language models provided by Lincoln Laboratory: 5,000-word, closed-vocabulary[3] and

3. A closed-vocabulary language model is intended for recognizing speech that does not include words outside of the vocabulary.

20,000-word open-vocabulary language models were used for the WSJ0 and WSJ1 experiments, respectively. The trigram language model was implemented using the N-best rescoring paradigm [19], by rescoring the list of the N-best sentence hypotheses generated using the bigram language model.

In the remainder of this section, we present results that show how mixture tying affects recognition performance. We also present experiments that investigate other modeling aspects of continuous HMMs, including modeling multiple vs. single observation streams and modeling time-correlation using linear discriminant analysis.

4.1 Degree of Mixture Tying

To determine the effect of mixture tying on the recognition performance, we evaluated a number of different systems on both WSJ0 and WSJ1. Table 1 compares the performance and the number of free parameters of tied mixtures, phonetically tied mixtures, and genonic mixtures on a development set that consists of 18 male speakers and 360 sentences of the 5,000-word WSJ0 task. The training data for this experiment included 3,500 sentences from 42 speakers. We can see that systems with a smaller degree of tying outperform the conventional tied mixtures by 25%, and at the same time have a smaller number of free parameters because of the reduction in the codebook size.

The difference in recognition performance between PTM and genonic HMMs is, however, much more dramatic in the WSJ1 portion of the database. There, the training data consisted of 37,000 sentences from 280 speakers, and gender-dependent models were built. The male subset of the 20,000-word, November 1992 evaluation set was used, with a bigram language model. Table 2 compares various degrees of tying by varying the number of genones used in the system. We can see that, because of the larger amount of available training data, the improvement in performance of genonic systems over PTM systems is much larger (20%) than in our 5,000-word experiments. Moreover, the best performance is achieved for a larger number of genones—1,700 instead of the 495 used in the 5,000-word experiments. These results are depicted in Figure 2.

In Table 3 we explore the additional degree of freedom that genonic HMMs have over fully continuous HMMs, namely that states mapped to the same genone can have different mixture weights. We can see that tying the mixture weights in addition to the Gaussians introduces a significant degradation in recognition performance. This degradation increases when the features are modeled using multiple observation streams (see Section 4.2) and as the amount of training data and the number of genones decrease.

4.2 Multiple vs. Single Observation Streams

Another traditional difference between fully continuous and tied mixture systems is the independence assumption of the latter when modeling multiple speech features. Tied mixture systems typically model static and dynamic spectral and energy features as conditionally independent observation streams, given the HMM state. The reason is that tied mixture systems provide a very coarse representation of the acoustic space, which makes it necessary to quantize each feature separately and artificially increase the resolution by modeling the features as independent. Then, the number of bins of the augmented feature is equal to the product of the number of bins of all individual features. The disadvantage is, of course, the independence assumption. When, however, the degree of tying is smaller, the finer representation of the acoustic space makes it unnecessary to improve the resolution accuracy by modeling the features as independent, and the feature-independence assumption can be removed. This claim is verified experimentally in Table 4. The first row in Table 4 shows the recognition performance of a system that models the six static and dynamic spectral and energy features as independent observation streams. The second row shows the performance of a system that models the six features in a single stream. We can see that the performance of the two systems is similar.

4.3 Linear Discriminant Features

For a given HMM state sequence, the observed features at nearby frames are highly correlated. HMMs, however, model these observations as conditionally independent, given the underlying state sequence. To capture local time correlation, we used a technique similar to the one described in [10]. Specifically, we used a linear discriminant feature extracted

using a linear transformation of the vector consisting of the cepstral and energy features within a window centered around the current analysis frame. The discriminant transformation was obtained using linear discriminant analysis [11] with classes defined as the HMM state of the context-independent phone. The state index assigned to each frame was determined using the maximum *a-posteriori* criterion and the forward-backward algorithm.

We found that the performance of the linear discriminant feature was similar to that of the original features, and that performance improves if the discriminant feature vector is used in parallel with the original cepstral features as a separate observation stream. From Table 5, we can see that the linear discriminant feature reduced the error rate on the WSJ1 20,000-word open-vocabulary male development set by approximately 7% using either a bigram or a trigram language model.

The best-performing system with 1,700 genones and the linear discriminant feature was then evaluated on various test and development sets of the WSJ database using bigram and trigram language models. Our word recognition results, summarized in Table 6, are comparable to the best reported results to date on these test sets [3][4][5].

5 REDUCING GAUSSIAN COMPUTATIONS

Genonic HMM recognition systems require evaluation of very large numbers of Gaussian distributions, and can be very slow during recognition. In this section, we will show how to reduce this computation while maintaining recognition accuracy. For simplicity, we use a baseline system in this section that has 589 genones, each with 48 Gaussian distributions, for a total of 28,272 39-dimensional Gaussians. This system has a smaller number of genones than the best-performing system of Section 4 and no context-dependent modeling across words. It runs much faster than our most accurate system, but its performance of 13.4% word error on ARPA's November 1992, 20,000-word evaluation test set using a bigram language model is slightly worse than our best result of 11.4% on this test set when the linear discriminant feature is not used (Table 2). Decoding time from word lattices is 12.2 times slower than real time on an R4400 processor. Full-grammar decoding time would be much slower.[4] Since the decoding time of a genonic recognition system such as

this one is dominated by Gaussian evaluation, reducing the number of Gaussians that require evaluation at each frame is critical for both fast experimentation and practical applications of the technology. We have explored two methods of reducing Gaussian computation: Gaussian clustering and Gaussian shortlists.

5.1 Gaussian Clustering

The number of Gaussians per genone can be reduced using clustering. Specifically, we used an agglomerative procedure to cluster the component densities within each genone to a smaller number. We considered several criteria that were used in [20], like an entropy-based and a generalized likelihood-based distortion measure. We found that the entropy-based measure worked better. This criterion is the continuous-density analog of the increase in weighted-by-counts entropy of the discrete HMM mixture-weight distributions that we used in the agglomerative clustering step of the genonic HMM system construction. Specifically, the cost of pooling two Gaussian densities—$N_i(x_t\,;\,\mu_i, \Sigma_i)$ and $N_j(x_t\,;\,\mu_j, \Sigma_j)$ —is the difference between the entropy of the pooled Gaussian and the sum of the entropies of the initial densities, all weighted by the number of samples used to estimate each density:

$$d(i,j) = \frac{n_i + n_j}{2}\log|\Sigma_{i\cup j}| - \frac{n_i}{2}\log|\Sigma_i| - \frac{n_j}{2}\log|\Sigma_j| \quad . \tag{11}$$

where n_i, n_j are the number of samples used to estimate the initial densities and $N_{i\cup j}(x_t\,;\,\mu_{i\cup j}, \Sigma_{i\cup j})$ is the pooled density.

In Table 7 we can see that the number of Gaussians per genone can be reduced by a factor of three by first clustering and then performing one additional iteration of the Baum-Welch algorithm. The table also shows that clustering followed by additional training iterations gives better accuracy than directly training a system with a smaller number of Gaussians (Table 7, Baseline 2). This is especially true as the number of Gaussians per genone decreases.

---

4. In the remainder of this section, all decoding times are from word lattices.

5.2 Gaussian Shortlists

Although clustering reduces the total number of Gaussians significantly, all the Gaussians belonging to genones used by HMM states that are in the Viterbi beam search must be evaluated at each frame during recognition. This evaluation includes a large amount of redundant computation; we have verified experimentally that the majority of the Gaussians will yield negligible probabilities. As a result, after reducing the Gaussians by a factor of three using clustering, the decoding time from word lattices is still 7.9 times slower than real time.

We have developed a method similar to the one introduced by Bocchieri [21] for preventing a large number of unnecessary Gaussian computations. Our method is to partition the acoustic space and for each partition to build a *Gaussian shortlist,* a list which specifies the subset of the Gaussian distributions expected to have high likelihood values in a given region of the acoustic space. First, vector quantization (VQ) is used to subdivide the acoustic space into VQ regions. Then, one list of Gaussians is created for each combination of VQ region and genone. The lists are created empirically, by considering a sufficiently large amount of speech data. For each acoustic observation, each Gaussian distribution is evaluated. Those distributions whose likelihoods are within a predetermined fraction of the most likely Gaussian are added to the list for that VQ region and genone. This scheme will result in some empty or too short lists. We have found that empty lists can cause a degradation in recognition performance, which can be avoided by enforcing a minimum shortlist size—we add to empty shortlists those Gaussians of the genone that achieve the highest likelihood for some observations quantized to the VQ region.

When recognizing speech, each observation is vector quantized, and only those Gaussians which are found in the shortlist are evaluated. This technique has allowed us to reduce by more than a factor of five the number of Gaussians considered each frame when applied to unclustered genonic recognition systems. Here we apply Gaussian shortlists to the clus-

tered system described in Section 5.1. Several methods for generating improved, smaller Gaussian shortlists are discussed and applied to the same system.

Table 8 shows the word error rates for shortlists generated by a variety of methods. Through these methods, we reduced the average number of Gaussian distributions evaluated for each genone from 18 to 2.48 without compromising accuracy. The various shortlists tested were generated in the following ways:

- **None**: No shortlist was used. This is the baseline case from the clustered system described above. All 18 Gaussians are evaluated whenever a genone is active.
- **12D-256**: To partition the acoustic space, the vector of 12 cepstral coefficients is quantized using a VQ codebook with 256 codewords. With unclustered systems, this method generally achieves a 5:1 reduction in Gaussian computation. In this clustered system, only a 3:1 reduction was achieved, most likely because the savings from clustering and Gaussian shortlists overlap. The average shortlist length was 6.08.
- **39D-256**: The cepstral codebook that partitions the acoustic space in the previous method ignores 27 of the 39 feature dimensions. By using a 39-dimensional, 256-codeword VQ codebook, we created better-differentiated acoustic regions and reduced the average shortlist length to 4.93.
- **39D-4096-min3**: We further decreased the number of Gaussians per region by shrinking the size of the regions. Here we used a single-feature VQ codebook with 4096 codewords, and reduced the average shortlist size to 3.68. For such a large codebook, vector quantization can be accelerated using a binary tree VQ fastmatch [22]. The minimum shortlist size was 3.
- **39D-4096-min1**: In our experiments with 48 Gaussians/genone, we found it important to ensure that each list contained a minimum of three Gaussian densities. With our current clustered systems we found that we can achieve similar recognition accuracy with a minimum shortlist size of one. As shown in Table 8, this technique results in lists with an average of 2.48 Gaussians per genone, without degradation in recognition accuracy.

Our results on the computational reduction on the evaluation of Gaussian likelihoods are summarized in Figure 3. We started with a speech recognition system with 48 Gaussians per genone (a total of 28,272 Gaussian distributions) that evaluated 14,538 Gaussian likelihood scores per frame and achieved a 13.4% word error rate running 12.2 times slower than real time on word lattices. Combining the clustering and Gaussian shortlist techniques described in Section 5, we managed to decrease the average number of Gaussians contained in each list to 2.48. As a result, the system's computational requirements were reduced to 732 Gaussian evaluations per frame, resulting in a system with word error of 13.5%, running at 2.5 times real time from word lattices.

6 CONCLUSIONS

An algorithm has been developed that balances the trade-off between resolution and trainability. Our method generalizes the tying of mixture components in continuous HMMs and achieves the degree of tying that is best suited to the available training data and the size of the recognition problem that we have in hand. We demonstrated in the large-vocabulary WSJ database that by selecting the appropriate degree of tying, the word-error rate can be decreased by 25% over conventional tied-mixture HMMs. To cope with the increase in computational requirements compared to tied-mixture HMMs, we have presented fast algorithms for evaluating the likelihoods of Gaussian mixtures. The number of Gaussians evaluated per frame was reduced by a factor of 20 and the decoding time by a factor of 6.

REFERENCES

[1] X. D. Huang and M. A. Jack, "Performance Comparison Between Semi-continuous and Discrete Hidden Markov Models," *IEE Electronics Letters*, Vol. 24 No. 3, pp. 149-150.

[2] J. R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. ASSP*, Vol. 38(12), pp. 2033-2045, December 1990.

[3] D. Pallet, J. G. Fiscus, W. M. Fisher, and J. S. Garofolo, "1993 Benchmark Tests for the ARPA Spoken Language Program," *HLT Workshop*, Princeton, NJ, March 1994.

[4] J. L. Gauvain, L. F. Lamel, G. Adda, and M. Adda-Decker, "The LIMSI Continuous Speech Dictation System: Evaluation on the ARPA Wall Street Journal Task," *Proc. ICASSP*, pp. I-125 - I-128, April 1994.

[5] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large Vocabulary Continuous Speech Recognition Using HTK," *Proc. ICASSP*, pp. II-125 - II-128, April 1994.

[6] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities," *Bell Systems Tech. Journal*, Vol. 64(6), pp. 1211-34, 1985.

[7] O. Kimball and M. Ostendorf, "On the Use of Tied-Mixture Distributions," *Proc. ARPA HLT Workshop*, March 1993.

[8] D. B. Paul, "The Lincoln Robust Continuous Speech Recognizer," *Proc. ICASSP*, pp. 449-452, May 1989.

[9] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer Speech and Language*, pp. 127-165, April 1990.

[10] X. Aubert, R. Haeb-Umbach and H. Ney, "Continuous Mixture Densities and Linear Discriminant Analysis for Improved Context-Dependent Acoustic Models," *Proc. ICASSP*, pp. 648-651, April 1993.

[11] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, J. Wiley & Sons, 1973.

[12] K. F. Lee, "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition," *IEEE Trans. ASSP*, pp. 599-609, April 1990.

[13] M.-Y. Hwang and X. D. Huang, "Subphonetic Modeling with Markov States - Senone," *Proc. ICASSP*, pp. I-33 - 36, March 1992.

[14] D. B. Paul and E. A. Martin, "Speaker Stress-resistant Continuous Speech Recognition," *Proc. ICASSP*, pp. 283-286, April 1988.

[15] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees," DARPA Workshop on Speech and Natural Language, pp. 264-269, February 1991.

[16] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes," *Inequalities*, Vol. 3, pp. 1-8, 1972.

[17] G. Doddington, "CSR Corpus Development," in *DARPA SLS Workshop*, Feb 1992.

[18] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large Vocabulary Dictation using SRI's DECIPHER™ Speech Recognition System: Progressive Search Techniques," *Proc. ICASSP*, pp. II-319 - II-322, April 1993.

[19] R. Schwartz and Y.-L. Chow, "A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses," *Proc. ICASSP*, pp. 701-704, May 1991.

[20] A. Kannan, M. Ostendorf, and J. R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," in *IEEE Trans. Speech and Audio Processing*, to appear July 1994.

[21] E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," *Proc. ICASSP*, pp. II-692 - II-695, April 1993.

[22] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proceedings of the IEEE*, Vol. 73, No. 11, pp. 1551-1588, November 1985.

TABLES

System	Number of Genones	Gaussians per genone	Total Parameters (thousands)	Word Error (%)
TM	1	256	5,126	14.1
PTM	40	100	2,096	11.6
Genones	495	48	1,530	10.6

TABLE 1. Comparison of various degrees of tying on a 5,000-word WSJ0 development set

	PTM	Genonic HMMs			
Number of Genones	40	760	1250	1700	2400
Word error rate (%)	14.7	12.3	11.8	11.4	12.0

TABLE 2. Recognition performance on the male subset of 20,000-word WSJ November 1992 ARPA evaluation set for various numbers of codebooks using a bigram language model

			Word Error (%)	
Recognition Task	Number of Genones	Number of Streams	Tied	Untied
5K WSJ0	495	6	9.7	7.7
20K WSJ1	1,700	1	12.2	11.4

TABLE 3. Comparison of state-specific vs. genone-specific mixture weights for different recognition tasks

System	Sub (%)	Del (%)	Ins (%)	Word Error (%)
6 streams	9.0	0.8	2.5	12.3
1 stream	8.7	0.8	2.3	11.8

TABLE 4. Comparison of modeling using 6 versus 1 observation streams for the 6 underlying features on the male subset of 20,000-word WSJ November 1992 evaluation set with a bigram language model

System	Bigram LM	Trigram LM
1,700 Genones	20.5	17.0
+ Linear Discriminants	19.1	15.8

TABLE 5. Word error rates (%) on the 20,000-word open-vocabulary male development set of the WSJ1 corpus with and without linear discriminant transformations

	Test set		
Grammar	Nov92	WSJ1 Dev	Nov93
Bigram	11.2	16.6	16.2
Trigram	9.3	13.6	13.6

TABLE 6. Word error rates on the November 1992 evaluation, the WSJ1 development, and the November 1993 evaluation sets using 20,000-word open-vocabulary bigram and trigram language models

System	Gaussians per Genone	Word Error (%)
Baseline1	48	13.4
Baseline1+Clustering	18	14.2
above+Retraining	18	13.6
Baseline2	25	14.4

TABLE 7. Improved training of systems with fewer Gaussians by clustering from a larger number of Gaussians

Shortlist	Shortlist Length	Gaussians Evaluated per Frame	Word Error (%)
none	18	5459	13.6
12D-256	6.08	1964	13.5
39D-256	4.93	1449	13.5
39D-4096-min3	3.68	1088	13.6
39D-4096-min1	2.48	732	13.5

TABLE 8. Word error rates and Gaussians evaluated, for a variety of Gaussian shortlists

FIGURES

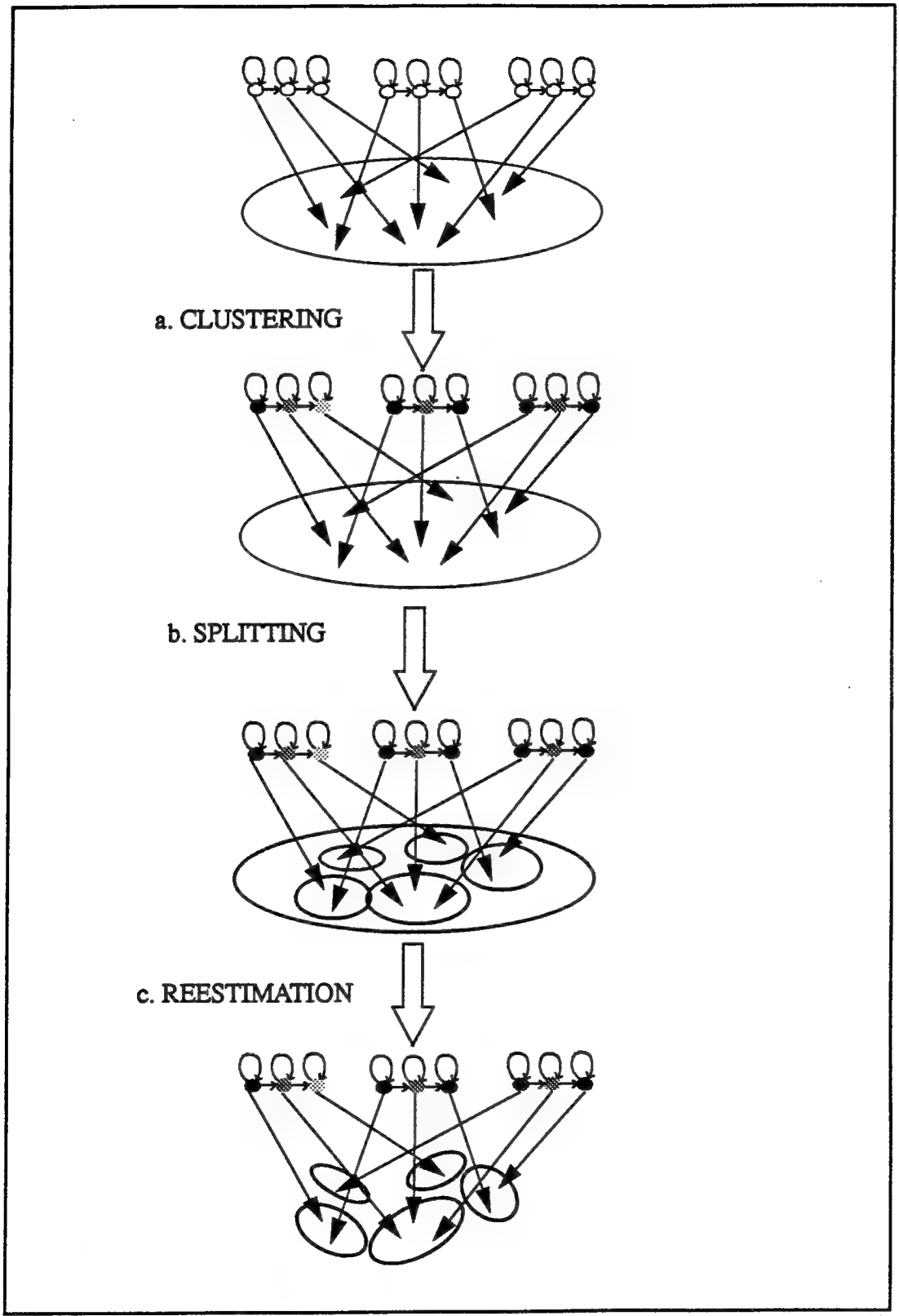

FIGURE 1. Construction of genonic mixtures. Arrows represent the stochastic mappings from state to mixture component. Ellipses represent the sets of Gaussians in a single genone.

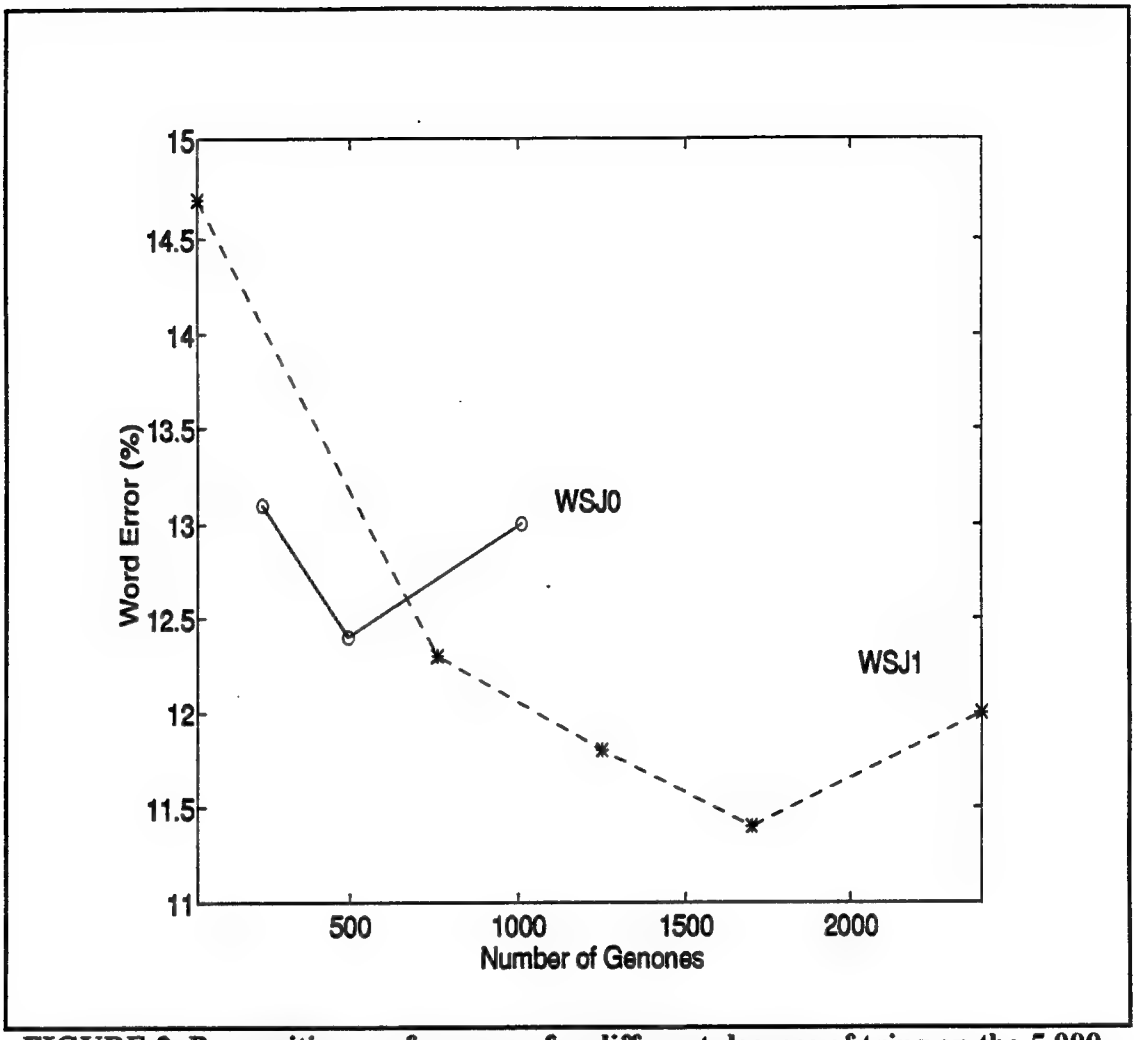

FIGURE 2. Recognition performance for different degrees of tying on the 5,000-word WSJ0 and 20,000-word WSJ1 tasks of the WSJ corpus.

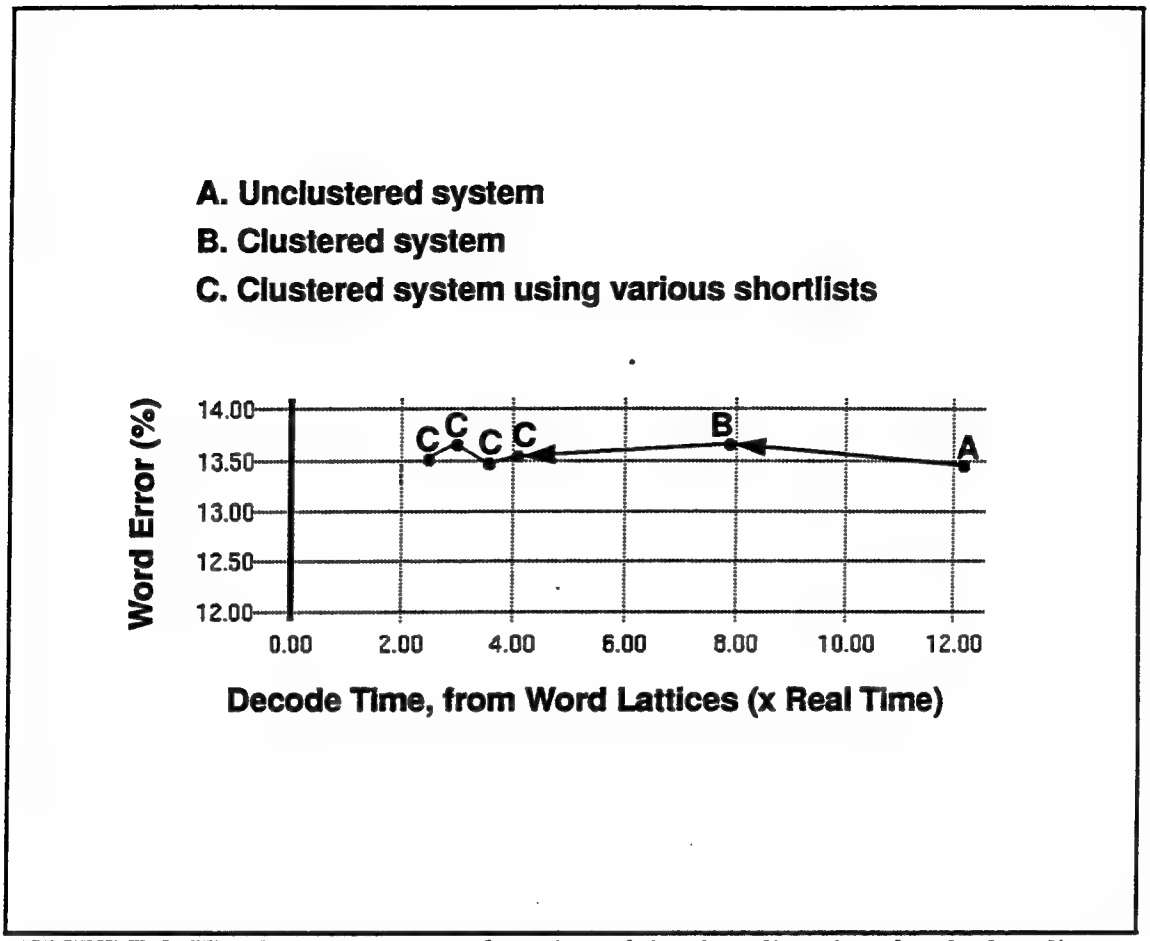

FIGURE 3. Word error rate as a function of the decoding time for the baseline system (A) and systems with fast Gaussian evaluation schemes (B and C).

HIGH-ACCURACY LARGE-VOCABULARY SPEECH RECOGNITION USING MIXTURE TYING AND CONSISTENCY MODELING

Vassilios Digalakis and Hy Murveit

SRI International
Speech Technology and Research Laboratory
333 Ravenswood Ave., Menlo Park, CA 94025-3493

ABSTRACT

Improved acoustic modeling can significantly decrease the error rate in large-vocabulary speech recognition. Our approach to the problem is twofold. We first propose a scheme that optimizes the degree of mixture tying for a given amount of training data and computational resources. Experimental results on the Wall Street Journal (WSJ) Corpus show that this new form of output distribution achieves a 25% reduction in error rate over typical tied-mixture systems. We then show that an additional improvement can be achieved by modeling local time correlation with linear discriminant features.

1. INTRODUCTION

To improve the acoustic-modeling component of SRI's DECIPHER™ speech recognition system, our research has focused on two main directions. The first is to decrease the degree of mixture tying in the mixture observation densities, since continuous-density hidden Markov models (HMMs) have recently been shown to outperform discrete-density and tied-mixture HMMs [16]. The second is the removal of the simplifying output independence assumption commonly used in HMMs.

Tied mixtures (TM) achieve robust estimation and efficient computation of the density likelihoods. However, the typical mixture size used in TM systems is small and does not provide a good representation of the acoustic space. Increasing the number of the mixture components (the codebook size) is not a feasible solution, since the mixture-weight distributions become too sparse. In large-vocabulary problems, where a large number of basic HMMs is used and each has only a few observations in the training data, sparse mixture-weight distributions cannot be estimated robustly and are expensive to store. To solve this problem, we follow the approach of simultaneously reducing the codebook size and increasing the number of different sets of mixture components (or codebooks). This procedure reduces the degree of tying, and the two changes can be balanced so that the total number of component densities in the system is effectively increased. The mapping from HMM states to codebooks can be determined using clustering techniques. Since our algorithm transforms a "less" continuous, or tied-mixture system, to a "more" continuous one, it has enabled us to investigate a number of traditional differences between tied-mixture and fully continuous HMMs, including codebook size and modeling of the speech features using multiple vs. single observation streams.

Our second main research direction is focused on removing the simplifying assumption used in HMMs that speech features from different frames are statistically independent given the underlying state sequence. In this paper we will deal with the modeling of the local temporal dependencies, that is, ones that span the duration of a phonetic segment. We will show through the use of recognition experiments and information theoretic criteria that achieving decorrelation of the speech features is not a sufficient condition for the improvement in recognition performance. To achieve the latter, it is necessary to improve the discrimination power of the output distributions through the use of new information. Local correlation modeling has recently been incorporated in our system through the use of linear discriminant features, and has reduced the word error rate by 7% on the Wall Street Journal (WSJ) corpus.

The remainder of the paper is organized as follows: in Section 2 we present the general form of mixture observation distributions used in HMMs, we discuss variations of this form that have appeared in the literature, and present an algorithm that enables us to adjust the mixture tying for optimum recognition performance. In Section 3 we deal with the problem of local time-correlation modeling: we comment on the potential improvement in recognition performance by incorporating conditional distributions, and describe the type of local consistency modeling currently used in our system. In Section 4 we present experimental results on the WSJ Corpus. These results are mainly a by-product of the system development for the November 1993 ARPA evaluation [16]. Finally, we conclude in Section 5.

2. GENONIC MIXTURES

A typical mixture observation distribution in an HMM-based speech recognizer has the form

$$p(x_t \mid s) = \sum_{q \in Q(s)} p(q|s) f(x_t|q) \qquad (1)$$

where s represents the HMM state, x_t the observed feature at frame t, and $Q(s)$ the set of mixture-component densities used in state s. We will use the term *codebook* to denote the set $Q(s)$. The stream of continuous vector observations can be modeled directly using Gaussians or other types of densities in the place

of $f(x_t \mid q)$, and HMMs with this form of observation distributions are known as continuous HMMs [19].

Various forms of tying have appeared in the literature. When tying is not used, the sets of component densities are different for different HMM states—that is, $Q(s) \neq Q(s')$ if $s \neq s'$. We will refer to HMMs that use no sharing of mixture components as *fully continuous* HMMs. The other extreme is when all HMM states share the same set of mixture components—that is, $Q(s) = Q$ is independent of the state s. HMMs with this degree of sharing were proposed in [8], [2] under the names *Semi-Continuous and Tied-Mixture* (TM) HMMs. Tied-mixture distributions have also been used with segment-based models, and a good review is given in [11]. Intermediate degrees of tying have also been examined. In phone-based tying, described in [17], [13], only HMM states that belong to allophones of the same phone share the same mixture components—that is, $Q(s) = Q(s')$ if s and s' are states of context-dependent HMMs with the same center phone. We will use the term *phonetically tied* to describe this kind of tying. Of course, for context-independent models, phonetically tied and fully continuous HMMs are equivalent. However, phonetically tied mixtures (PTM) did not significantly improve recognition performance in previous work.

The continuum between fully continuous and tied-mixture HMMs can be sampled at any other point. The choice of phonetically tied mixtures, although linguistically motivated, is somewhat arbitrary and may not achieve the optimum trade-off between resolution and trainability. We have recently introduced an algorithm [4] that allows as to select the degree of tying that attains optimum recognition performance for the given computational resources. This algorithm follows a bootstrap approach from a system that has a higher degree of tying (i.e., a TM or a PTM system), and progressively unties the mixtures using three steps: clustering, splitting and pruning, and reestimation.

2.1. Clustering

The HMM states of all allophones of a phone are clustered following an agglomerative procedure. The clustering is based on the weighted-by-counts entropy of the mixture-weight distributions [12]. The clustering procedure partitions the set of HMM states S into disjoint sets of states

$$S = S_1 \cup S_2 \cup \ldots \cup S_n \qquad (2)$$

The same codebooks will be used for all HMM states belonging to a particular cluster S_i.

2.2. Splitting and Pruning

After determination of the sets of HMM states that will share the same codebook, seed codebooks for each set of states that will be used by the next reestimation phase are constructed. These seed codebooks can be constructed by either one or a combination of two procedures:

- Identifying the most likely subset of mixture components of the boot system for each cluster of HMM states S_i and using these subsets $Q(S_i) \subset Q(S)$ as seed codebooks for the next phase
- Copying the original codebook multiple times (one for each cluster of states) and performing one iteration of the Baum-Welch algorithm over the training data with the new tying scheme; the number of component densities in each codebook can then be reduced using clustering [10]

2.3. Reestimation

The parameters are reestimated using the Baum-Welch algorithm. This step allows the codebooks to deviate from the initial values and achieve a better approximation of the distributions.

We will refer to the Gaussian codebooks as *genones* and to the HMMs with arbitrary tying of Gaussian mixtures as *genonic* HMMs. Clustering of either phone or subphone units in HMMs has also been used in [18], [12], [1], [9]. Mixture-weight clustering of different HMM states can reduce the number of free parameters in the system and, potentially, improve recognition performance because of the more robust estimation. It cannot, however, improve the resolution with which the acoustic space is represented, since the total number of component densities in the system remains the same. In our approach, we use clustering to identify sets of subphonetic regions that will share mixture components. The later steps of the algorithm, where the original set of mixture components is split into multiple overlapping genones and each one is reestimated using data from the states belonging to the corresponding cluster, effectively increase the number of distinct densities in the system and provide the desired detail in the resolution.

Reestimation of the parameters can be achieved using the standard Baum-Welch reestimation formulae for HMMs with Gaussian mixture observation densities, since tying does not alter their form, as pointed out in [21]. During recognition, and to reduce the large amount of computation involved in evaluating Gaussian likelihoods, we can use the fast computational techniques described in [15].

In place of the component densities $f(x_t \mid q)$ we use exponentially weighted Gaussian distributions:

$$p(x_t|s) = \sum_{q \in Q(s)} p(q|s) \left[N(x_t; \mu_q, \Sigma_q)\right]^{\alpha} \qquad (3)$$

where the exponent $\alpha \leq 1$ is used to reduce the dynamic range of the Gaussian scores (that would, otherwise, dominate the mixture probabilities $p(q \mid s)$) and also to provide a smoothing effect at the tails of the Gaussians.

3. TIME CORRELATION MODELING

For a given HMM state sequence, the observed features at nearby frames are highly correlated. Modeling time correlation can significantly improve speech recognition performance for two reasons. First, dynamic information is very important [6], and explicit time-correlation modeling can potentially outperform more traditional and simplistic approaches like the incorporation of cepstral derivatives as additional feature streams.

Second, sources of variability—such as microphone, vocal tract shape, speaker dialect, and speech rate—will not dominate the likelihood computation during Viterbi decoding by being rescored at every frame. We will call techniques that model such temporal dependencies *consistency modeling*.

The output-independence assumption is not necessary for the development of the HMM recognition (Viterbi) and training (Baum-Welch) algorithms. Both of these algorithms can be modified to cover the case when the features depend not only on the current HMM state, but also on features at previous frames [20]. However, with the exception of the work reported in [3] that was based on segment models, explicit time-correlation modeling has not improved the performance of HMM-based speech recognizers.

To investigate these results, we conducted a pilot study to estimate the potential improvement in recognition performance when using explicit correlation modeling over more traditional methods like time-derivative information. We used information-theoretic criteria and measured the amount of mutual information between the current HMM state and the cepstral coefficients at a previous "history" frame. The mutual information was always conditioned on the identity of the left phone, and was measured under three different conditions:

- $I(h,s)$—mutual information between the current HMM state s and a cepstral coefficient h at the history frame; a single, left-context-dependent Gaussian distribution for the cepstral coefficient at the history frame was hypothesized,
- $I(h,s|c)$—conditional mutual information between the current HMM state s and a cepstral coefficient h at the history frame when the corresponding cepstral coefficient c of the current frame is given; a left-context-dependent, joint Gaussian distribution for the cepstral coefficients at the current and the history frames was hypothesized,
- $I(h,s|c,d)$—same as above, but conditioned on both the cepstral coefficient c and its corresponding derivative d at the current frame.

The results are summarized in Table 1 for history frames with lags of 1, 2, 4 and a variable one. In the latter case, we condition the mutual information on features extracted at the last frame t_0 of the previous HMM state, as located by a forced Viterbi alignment. We can see from this table that in the unconditional case, the cepstral coefficients at frames closer to the current one provide more information about the identity of the current phone. However, the amount of additional information that these coefficients provide when the knowledge of the current cepstra and their derivatives is taken into account is smaller. The additional information in this case is larger for lags greater than 1, and is maximum for the variable lag.

These measurements predict that the previous frame's observation is not the optimal frame to use when conditioning a state's output distribution. To verify this, and to actually evaluate recognition performance, we incorporated time-correlation modeling in an HMM system with genonic mixtures. Specifically, we generalized the Gaussian mixtures to mixtures of conditional Gaussians, with the current cepstral coefficient x_t conditioned on the corresponding cepstral coefficient x_{t_0} of the history frame t_0:

Lag t_0	0	1	2	4	Variable
$I(h, s)$	0.28	0.27	0.25	0.19	0.25
$I(h, s \mid c)$	0	0.13	0.15	0.15	0.21
$I(h, s \mid c, d)$	0	0.11	0.14	0.13	0.20

Table 1. Mutual information (in bits) between HMM state s at time t and cepstral coefficient h at time t-t_0 for various lags; included is the conditional mutual information when the corresponding cepstral coefficient and its derivative at time t are given

$$p\left(x_t \mid s, x_{t-t_0}\right) = \sum_{q \in Q(s)} p(q|s) f\left(x_t | q, x_{t_0}\right) \qquad (4)$$

We either replaced the original unconditional distributions of the cepstral coefficients and their derivatives with the conditional Gaussian distributions, or we used them in parallel as additional observation streams. The results on the 5,000-word recognition task of the WSJ0 corpus are summarized in Table 2 for fixed-lag history frames. We can see that the recognition results are in perfect agreement with the behavior predicted by the mutual-information study. The improvements in recognition performance over the system that does not use conditional distributions are actually proportional to the measured amount of conditional mutual information at the various history frames. However, these improvements are small and statistically insignificant, and indicate that the derivative features effectively model the local dynamics.

Delay	Word Error—Conditional only (%)	Word Error—Both (%)	$I(h, s \mid c, d)$
0	10.32	-	0
1	10.98	10.19	0.11
2	10.50	9.65	0.14
4	10.32	9.83	0.13

Table 2. Recognition rates on 5,000-word WSJ corpus with conditional distributions either replacing the unconditional ones or used in parallel

Instead of using conditional Gaussian distributions, one can alternatively choose to use features obtained with linear discriminants. Local time correlation can be modeled by estimating the transformations over multiple consecutive frames [5],[7]. This approach has the additional advantage that it is computationally less expensive, since the discriminant transformations can be computed in the recognizer front end and only once at each frame. However, as we will see in the following section, linear discriminants gave only moderate improvements in recognition performance, and this is consistent with the conditional Gaussian results of this section. From the conditional information measurements that we have presented, we can see that in order to provide additional information to the recognizer we must condition the output distributions not only on a previous history frame, but also on the start time of the current subphonetic segment, and this is an area that we are currently investigating.

4. EXPERIMENTAL RESULTS

We used the algorithms described in this paper on the 5,000- and 64,000-word recognition tasks of the WSJ corpus. We used the progressive-search framework [14] for fast experimentation. With this approach, an initial fast recognition pass creates word lattices for all sentences in the development set. These word lattices are used to constrain the search space in all subsequent experiments. In our development we used both the WSJ0 5,000 word and the WSJ1 64,000 word portions of the database, and the baseline bigram and trigram language models provided by Lincoln Laboratory.

4.1. Degree of Mixture Tying

To determine the effect of mixture tying on the recognition performance, we evaluated a number of different systems on both WSJ0 and WSJ1. Table 3 compares the performance and the number of free parameters of tied mixtures, phonetically tied mixtures, and genonic mixtures on a development set that consists of 18 male speakers and 360 sentences of the 5,000-word WSJ0 task. The training data for this experiment included 3,500 sentences from 42 speakers. We can see that systems with a smaller degree of tying outperform the conventional tied mixtures by 25%, and at the same time have a smaller number of free parameters because of the reduction in the codebook size.

System	Number of Genones	Gaussians per genone	Total Parameters (thousands)	Word Error (%)
TM	1	256	5,126	14.1
PTM	40	100	2,096	11.6
Genones	495	48	1,530	10.6

Table 3. Comparison of various degrees of tying on 5,000-word WSJ development set

The difference in recognition performance between PTM and genonic HMMs with smaller tying is, however, much more dramatic in the WSJ1 portion of the database. The training data consisted of 37,000 sentences from 280 speakers, and gender-dependent models were built. The male subset of the 20,000-word November 1992 evaluation set was used, with a bigram language model. Table 4 compares various degrees of tying by varying the number of genones used in the system. We can see that, because of the larger amount of available training data, the improvement in performance of genonic systems over PTM systems is much larger (20%) than in our 5,000-word experiments. Moreover, the best performance is achieved for a larger number of genones—1,700 instead of the 495 used in the 5,000-word experiments. These results are depicted in Figure 1.

	PTM	Genonic HMMs			
Number of Genones	40	760	1250	1700	2400
Word error rate (%)	14.7	12.3	11.8	11.4	12.0

Table 4. Recognition performance on the male subset of 20,000-word WSJ November 1992 ARPA evaluation set for various numbers of codebooks using a bigram language model.

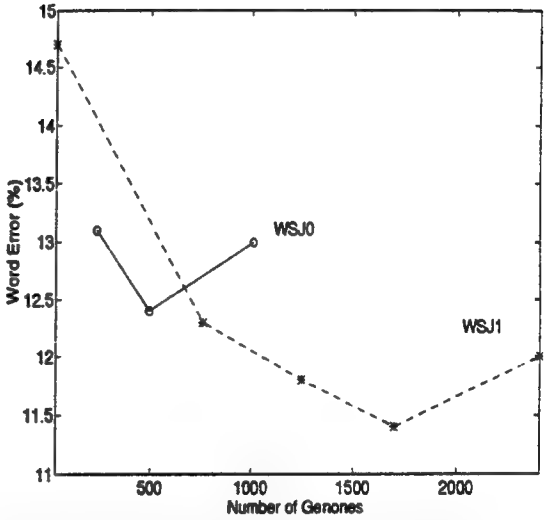

Figure 1: Recognition performance for different degrees of tying on the 5,000-word WSJ0 and 20,000-word WSJ1 tasks of the WSJ corpus

In Table 5 we explore the additional degree of freedom that genonic HMMs have over fully continuous HMMs, namely that states mapped to the same genone can have different mixture weights. We can see that tying the mixture weights in addition to the Gaussians introduces a significant degradation in recognition performance. This degradation increases when the features are modeled using multiple observation streams (see following section) and as the amount of training data and the number of genones decrease.

	Number of Genones	Number of Streams	Word Error (%) Tied	Untied
5K WSJ0	495	6	9.7	7.7
20K WSJ1	1,700	1	12.2	11.4

Table 5. Comparison of state-specific vs. genone-specific mixture weights for different recognition tasks

4.2. Multiple vs. Single Observation Streams

Another traditional difference between fully continuous and tied mixture systems is the independence assumption of the latter when modeling multiple speech features. Tied mixture systems typically model static and dynamic spectral and energy features as conditionally independent observation streams given the HMM state, because tied mixture systems provide a very coarse representation of the acoustic space. It is, therefore, necessary to "quantize" each feature separately and artificially increase the resolution by modeling the features as independent: the number of "bins" of the augmented feature is equal to the product of the number of "bins" of all individual features. The disadvantage is, of course, the independence assumption. When, however, the degree of tying is smaller, the finer representation of the acoustic space makes it unnecessary to artificially improve the resolution accuracy by modeling the features as independent. Hence, for systems that are loosely tied we can remove the feature-independence assumption. This claim is verified experimentally in Table 6. The first row shows the recognition performance of a system that models the six static and dynamic spectral and energy features used in DECIPHER™ as independent observation streams. The second row shows the performance of a system that models the six features in a single stream. We can see that the performance of the two systems is similar.

System	Sub (%)	Del (%)	Ins (%)	Word Error (%)
6 streams	9.0	0.8	2.5	12.3
1 stream	8.7	0.8	2.3	11.8

Table 6. Comparison of modeling using 6 versus 1 observation streams for 6 underlying features on the male subset of 20,000-word WSJ November 1992 evaluation set with a bigram language model

4.3. Linear Discriminant Features

To capture local time correlation we used a linear discriminant feature extracted using a transformation of the features within a window around the current frame. The discriminant transformation was obtained using linear discriminant analysis with classes defined as the HMM state of the context-independent phone. The state index that was assigned to the frame was determined using the maximum *a-posteriori* criterion and the forward-backward algorithm.

We found that the performance of the linear discriminant feature was similar to that of the original features. However, we found that an improvement in performance can be obtained if the discriminant features are used in parallel with the original features. A genonic HMM system with 1,700 genones and linear discriminants as an additional feature was evaluated on the 20,000-word open-vocabulary November 1993 ARPA evaluation set. It achieved word-error rates of 16.5% and 14.5% with the standard bigram and trigram language models, respectively. These results, however, were contaminated by the presence of a large DC offset in most of the waveforms of the phase 1 WSJ1 corpus. We later removed the DC offset from the waveforms, and reestimated the models using the exact procedure followed during the development of the system used in the November 1993 evaluation. From Table 6, we can see that the linear discriminant feature reduced

System	Bigram LM	Trigram LM
1,700 Genones	20.5	17.0
+ Linear Discriminants	19.1	15.8

Table 7. Word error rates (%) on the 20,000-word open-vocabulary male development set of the WSJ1 corpus with and without linear discriminant transformations

the error rate on the WSJ1 20,000-word open-vocabulary male development set by approximately 7% using either a bigram or a trigram language model. Table 4 presents the results of the system with linear discriminants on various test and development sets.

	Test set		
Grammar	Nov92	WSJ1 Dev	Nov93
Bigram	11.2	16.6	16.2
Trigram	9.3	13.6	13.6

Table 8. Word error rates on the November 1992 evaluation, the WSJ1 development, and the November 1993 evaluation sets using 20,000-word open-vocabulary bigram and trigram language models

5. CONCLUSIONS

New acoustic modeling techniques significantly decrease the error rate in large-vocabulary continuous speech recognition. The genonic HMMs balance the trade-off between resolution and trainability, and achieve the degree of tying that is best suited to the available training data and computational resources. For example, one can decrease the computational load by decreasing the number of genones (i.e., increasing the degree of tying) with a small penalty in recognition performance [15]. Our results on the various test sets represent state-of-the-art recognition performance on the 20,000-word open-vocabulary WSJ task.

ACKNOWLEDGMENTS

We gratefully acknowledge support for this work from ARPA through Office of Naval Research Contract N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Government funding agencies.

REFERENCES

1. L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo and M. A. Picheny, "Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees,"

DARPA Workshop on Speech and Natural Language, pp. 264-269, February 1991.

2. J. R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. ASSP*, Vol. 38(12), pp. 2033-2045, Dec. 1990.
3. V. Digalakis, J. R. Rohlicek and M. Ostendorf, "ML Estimation of a Stochastic Linear System with the EM Algorithm and its Application to Speech Recognition," *IEEE Trans. Speech and Audio Processing*, October 1993.
4. V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," to appear in *Proc. ICASSP*, 1994.
5. G. R. Doddington, "Phonetically Sensitive Discriminants for Improved Speech Recognition," *Proceedings ICASSP-89*, pp. 556-559.
6. S. Furui, "On the Role of Spectral Transition for Speech Perception," *Journal of the Acoustical Society of America*, vol. 80(4), pp. 1016-1025, October 1986.
7. R. Haeb-Umbach and H. Ney, "Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition," *Proc. ICASSP*, pp. I-13 - I-16, March 1992.
8. X. D. Huang and M. A. Jack, "Performance Comparison Between Semi-continuous and Discrete Hidden Markov Models," *IEE Electronics Letters*, Vol. 24 no. 3, pp. 149-150.
9. M.-Y. Hwang and X. D. Huang, "Subphonetic Modeling with Markov States - Senone," *Proc. ICASSP*, pp. I-33-36, March 1992.
10. A. Kannan, M. Ostendorf and J. R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," in *IEEE Trans. Speech and Audio Processing*, to appear July 1994.
11. O. Kimball and M. Ostendorf, "On the Use of Tied-Mixture Distributions," *Proc. ARPA HLT Workshop*, March 1993.
12. K. F. Lee, "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition," *IEEE Trans. ASSP*, pp. 599-609, April 1990.
13. C. Lee, L. Rabiner, R. Pieraccini and J. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition," *Computer Speech and Language*, April. 1990, pp. 127-165.
14. H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large Vocabulary Dictation using SRI's DECIPHER™ Speech Recognition System: Progressive Search Techniques," *Proc. ICASSP*, pp. II-319 - II-322, April 1993.
15. H. Murveit, P. Monaco, V. Digalakis and J. Butzberger, "Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System," this proceedings.
16. D. Pallet, J. G. Fiscus, W. M. Fisher and J. S. Garofolo, "1993 Benchmark Tests for the ARPA Spoken Language Program," this proceedings.
17. D. B. Paul, "The Lincoln Robust Continuous Speech Recognizer," *Proc. ICASSP*, pp. 449-452, May 1989.
18. D. B. Paul and E. A. Martin, "Speaker Stress-resistant Continuous Speech Recognition," *Proc. ICASSP*, pp. 283-286, April 1988.
19. L. R. Rabiner, B. H. Juang, S. E. Levinson and M. M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities," *Bell Systems Tech. Journal*, Vol. 64(6), pp. 1211-34, 1985.
20. Wellekens, C., "Explicit Time Correlation in Hidden Markov Models for Speech Recognition," *Proc. ICASSP-87*.
21. S. J. Young, "The General Use of Tying in Phoneme-Based HMM Speech Recognizers," *Proc. ICASSP*, pp. I-569 - I-572, March 1992.

TECHNIQUES TO ACHIEVE AN ACCURATE REAL-TIME LARGE-VOCABULARY SPEECH RECOGNITION SYSTEM

Hy Murveit, Peter Monaco, Vassilios Digalakis, John Butzberger

SRI International
Speech Technology and Research Laboratory
333 Ravenswood Avenue
Menlo Park, California 94025-3493

ABSTRACT

In addressing the problem of achieving high-accuracy real-time speech recognition systems, we focus on recognizing speech from ARPA's 20,000-word Wall Street Journal (WSJ) task, using current UNIX workstations. We have found that our standard approach—using a narrow beam width in a Viterbi search for simple discrete-density hidden Markov models (HMMs)—works in real time with only very low accuracy. Our most accurate algorithms recognize speech many times slower than real time. Our (yet unattained) goal is to recognize speech in real time at or near full accuracy.

We describe the speed/accuracy trade-offs associated with several techniques used in a one-pass speech recognition framework:

- Trade-offs associated with reducing the acoustic modeling resolution of the HMMs (e.g., output-distribution type, number of parameters, cross-word modeling)
- Trade-offs associated with using lexicon trees, and techniques for implementing full and partial bigram grammars with those trees
- Computation of Gaussian probabilities are the most time-consuming aspect of our highest accuracy system, and techniques allowing us to reduce the number of Gaussian probabilities computed with little or no impact on speech recognition accuracy.

Our results show that tree-based modeling techniques used with appropriate acoustic modeling approaches achieve real-time performance on current UNIX workstations at about a 30% error rate for the WSJ task. The results also show that we can dramatically reduce the computational complexity of our more accurate but slower modeling alternatives so that they are near the speed necessary for real-time performance in a multipass search. Our near-future goal is to combine these two technologies so that real-time, high-accuracy large-vocabulary speech recognition can be achieved.

1. INTRODUCTION

Our techniques for achieving real-time, high-accuracy large-vocabulary continuous speech recognition systems focus on the task of recognizing speech from ARPA's Wall Street Journal (WSJ) speech corpus. All of the speed and performance data given in this paper are results of recognizing 169 sentences from the four male speakers that comprise ARPA's November 1992 20,000-word vocabulary evaluation set. Our best performance on these data is 8.9% (10.3% using bigram language models). Our standard implementation for this system would run approximately 100 times slower than real time.[1] Both these systems use beam-search techniques for finding the highest-scoring recognition hypothesis.

Our most accurate systems are those that use HMMs with genonic mixtures as observation distributions [3]. Genonic mixtures sample the continuum between fully continuous and tied-mixture HMMs at an arbitrary point and therefore can achieve an optimum recognition performance given the available training data and computational resources. In brief, genonic systems are similar to fully continuous Gaussian-mixture HMMs, except that instead of each state having its own set of Gaussian densities, states are clustered into *genones* that share these Gaussian codebooks. Each state, however, can have its own set of mixture weights used with the Gaussian codebook to form its own unique observation distribution. All the genonic systems discussed in this paper use a single 39-dimensional observation composed of the speech cepstrum and its first and second derivatives, and the speech energy and its first and second derivatives. All Gaussians have diagonal covariance matrices.

2. MODELING TRADE-OFFS

The speed/accuracy trade-off of our speech recognition systems can be adjusted in several ways. The standard approaches are to adjust the beam width of the Viterbi search and to change the

1. We define real-time systems as those that process 1 second of speech per second.

output-distribution modeling technique. Table 1 shows, for

System Type	Cross-Word Modeling	Word Error (%)	Lattice Search Speed
Genone	yes	11.6	50.4
Genone	no	13.4	19.8
Phonetically Tied Mixtures	yes	13.9	43.9
Phonetically Tied Mixtures	no	16.6	6.8
VQ	no	19.2	~1

Table 1: Effect of model type on speed and accuracy

instance, that eliminating cross-word modeling can significantly improve the speed of our recognition systems at about a 20% cost in word error. In this table, lattice speed refers to recognition accuracy when decoding from precomputed word lattices [8]. That is, this is only performing a subset of the search. Actual full grammar recognition time could be from a factor of 3 to an order of magnitude higher. However, it is useful to compare relative lattice decoding speeds from the various techniques.

A technique frequently used at SRI to achieve relatively fast speech recognition demonstrations is to downgrade our acoustic modeling by implementing a discrete density (VQ) HMM system without cross-word acoustic modeling. This system is then searched using a Viterbi search with a small beam width. Table 2 shows the full grammar speed accuracy trade-off when modifying the beam width if a Silicon Graphics Incorporated[2] (SGI) UNIX workstation with a 150-MHz MIPS R4400 CPU[3] is used to

Beam Width	Word Error (%)	Hypotheses per Frame	Full Search Speed
600	29.5	981	3.2
700	21.5	3089	8.3
800	19.2	7764	16.0

Table 2: Speed/accuracy trade-off for a beam search

perform the computation.

We have found that this technique gives an unsatisfactory speed/accuracy trade-off for this task and we have investigated other techniques as described below.

2. All product names mentioned in this paper are the trademark of their respective holders.

3. This workstation scores 85 and 93 for the SPECint92 and SPECfp92 benchmarks. For our tests it is roughly 50% faster than an SGI R4000 Indigo, and 50% faster than a SPARC 10/51. It should be between 1/2 and 2/3 the speed of an HP735. SGI R4400 systems cost about $12,000.

3. LEXICON TREES

We explored the use of lexicon trees as a technique for speeding up the decoding times for all modeling techniques. Lexicon trees represent the phonetics of the recognition vocabulary as a tree instead of as a list of pronunciations (lists of phones). With a tree representation, words starting with the same phonetic units share the computation of phonetic models. This technique has been used by others, including the IBM [10], Phillips [7], and CMU groups, and it is also currently used at LIMSI. Because of the large amount of sharing, trees can drastically reduce the amount of computation required by a speech recognition system. However, lexicon trees have several possible drawbacks:

- Phonetic trees are not able to use triphone modeling in all positions since the right phonetic context of a node in a tree can be ambiguous.
- One cannot implement an admissible Viterbi search for a single lexicon tree when using a bigram language model, because the word being decoded (*w2* in the bigram equation $P(w2/w1)$) may not be known until a leaf in the tree is reached—long after certain Viterbi decisions are typically made.

The first concern can be addressed by replicating nodes in the tree to disambiguate triphone contexts. However, even this may not be necessary because the large majority of right contexts in the tree are unambiguous (that is, most nodes have only one child). This is shown in Table 3, where the concentrations of triphone and biphone models are compared for tree- and linear-lexicon schemes.

Lexicon Type	Triphone Models (%)	Biphone Models (%)
Tree	73	27
Linear	85	15

Table 3: Model allocation for the SRI WSJ system with and without lexicon trees

The second concern, the ability to model bigram language models using an admissible search strategy, is a problem. As shown in Table 4, moving from a bigram to a unigram language model more than doubles our error rate. Ney [7] has proposed a scheme where lexicon trees are replicated for each bigram context. It is possible that this scheme would generalize to our application as well. For the three recognition systems in Table 2, on average 7, 13, and 26 different words end each frame. This is the minimum average number of copies of the lexicon tree that the system would need to maintain.

We have decided to pursue a different approach, which is shown in the figure below. We refer to this technique as *approximate bigram trees.*

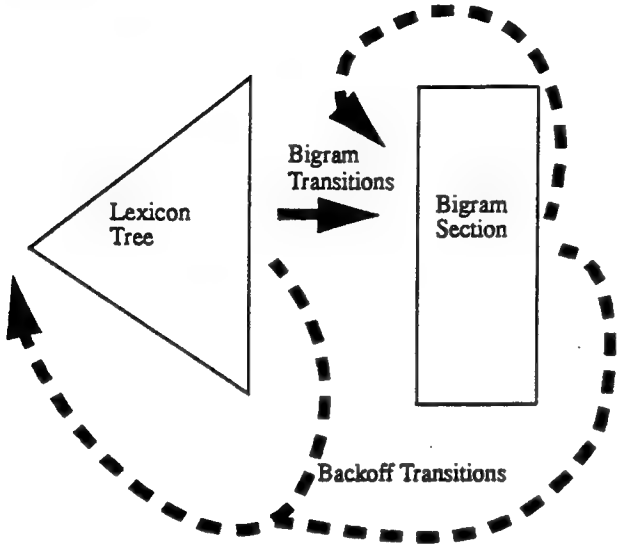

Figure 1: Approximate bigram trees

In an approximate bigram tree, the aim is to model the salient portion of the backed-off bigram language model [11] in use. In an approximate bigram tree, a standard lexicon tree (incorporating unigram word probabilities) is combined with a bigram section that maintains a linear (non-tree) representation of the vocabulary. Bigram and backoff language model transitions are added to the leaves of the tree and to the word-final nodes of the bigram section.[4] When the entire set of bigram is represented, then this network implements a full backed-off bigram language model with an efficient tree-based backoff section. In fact, for VQHMM systems, this scheme halves our typical decoding time for little or no cost in accuracy. Typically, however, we need further reduction in the computational requirement. To achieve this we represent only a subset of the group of bigram transitions (and adjust the backoff probabilities appropriately). This degrades the accuracy of our original bigram language model, but reduces its computational requirements. The choice of which bigrams to represent is the key design decision for approximate bigram trees. We have experimented with four techniques for choosing bigram subsets to see which make the best speed/accuracy trade-offs:

Count x means only use bigrams where $P(w1) * P(w2/w1) > 10^x$.

Prob x means only use bigrams where $P(w2/w1) > 10^x$.

Improve X means only use bigrams where $P(w2/w1) > Backoff(w1) * P(w2) / 10^x$.

Top *x* means only use bigrams $P(w2/w1)$ where *w2* is one of the most frequent *x* words.

[4] In the actual implementation, word-final nodes in the bigram section are merged with their counterparts in the tree so that the bigram transitions need be represented only once. For simplicity, however, we show the system with two sets of bigram probabilities.

Table 4 shows speed/accuracy trade-offs for approximate bigram trees.

Tree Type	Number of Bigrams Used (thousands)	Word Error (%)	Full Search Speed (x RT)
Unigram tree	0	42.3	0.6
(non-tree) Bigram	3500	21.5	8.5
count, -6	93	30.4	1.5
count, -5	10	35.8	0.9
count, -4	.6	39.2	0.7
prob, -3	1250	28.2	0.9
prob, -2.5	671	29.2	0.8
prob, -2	219	31.5	0.7
prob, -1	20	36.6	0.7
improve, 2	908	29.7	1.6
improve, 3	191	37.1	0.8
top 10	113	39.5	0.7
top 50	320	36.0	0.7
top 100		35.2	0.7
top 1000	1500	31.4	1.1
top 5000	2624	25.3	1.9
top 20000	3500	21.0	~3

Table 4: Performance of "approximate bigram" trees

The top two lines of the table show that the bigram language model improves performance from 42.3% word error to 21.5% as compared with a unigram language model. The rest of the table shows how approximate bigram trees can trade off the performance and speed of the bigram model. For instance, in several techniques shown—such as *prob 2.5*—that maintain more than half of the benefit of bigrams for little computational cost, CPU usage goes from 0.6 to 0.8, when the error rate goes from 42.3% to 29.2%. The rest of the improvement, reducing the error rate from 29.2% to 21%, increases the required computation rate by a factor of four.

Table 4 also shows that the number of bigrams represented does not predict the computation rate.

The square root of the perplexity of these language models seems to predict the recognition performance as shown in Table 5.

Top X	Perplexity	Perplexity Square Root	Word error (%)
0	1248	35.3	42.3
10	954	30.9	39.5
50	727	27.0	36.0
100	631	25.1	35.2
1000	401	20.0	31.4
20000	237	15.4	21

Table 5: Grammar Perplexity for top X trees

4. REDUCING GAUSSIAN COMPUTATIONS

SRI's most accurate recognition systems, using genonic mixtures, require the evaluation of very large numbers of Gaussian distributions, and are therefore very slow to compute. The baseline system referenced here uses 589 genonic mixtures (genones), each with 48 Gaussian distributions, for a total of 28,272 39-dimensional Gaussians. On ARPA's November 1992 20,000-word Evaluation Test Set, this noncrossword, bigram system performs at 13.43% word error. Decoding time from word lattices is 12.2 times slower than real time on an R4400 processor. Full grammar decoding time would be much slower. Since the decoding time of a genonic recognition system such as this one is dominated by Gaussian evaluation, one major thrust of our effort to achieve real-time recognition has been to reduce the number of Gaussians requiring evaluation each frame. We have explored three methods of reducing Gaussian computation: Gaussian clustering, Gaussian shortlists, and genonic approximations.

4.1. Gaussian Clustering

The number of Gaussians per genone can be reduced using clustering. Specifically, we used an agglomerative procedure to cluster the component densities within each genone. The criteria that we considered were an entropy-based distance and a generalized likelihood-based distance [6]. We found that the entropy-based distance worked better. This criterion is the continuous-density analog of the weighted-by-counts entropy of the discrete HMM state distributions, often used for clustering HMM state distributions [5], [3].

Our experiments showed that the number of Gaussians per genone can be reduced by a factor of three by first clustering and then performing one additional iteration of the Baum-Welch algorithm as shown in Table 6. The table also shows that clustering followed by additional training iterations gives better accuracy than directly training a system with a smaller number of Gaussians (Table 6, Baseline2). This is especially true as the number of Gaussians per genone decreases.

System	Gaussians per Genone	Word Error (%)
Baseline1	48	13.43
Baseline1+Clustering	18	14.17
above+Retraining	18	13.64
Baseline2	25	14.35

Table 6: Improved training of systems with fewer Gaussians by clustering from a larger number of Gaussians

4.2. Gaussian Shortlists

We have developed a method for eliminating large numbers of Gaussians before they are computed. Our method is to build a "Gaussian shortlist" [2], [4], which uses vector quantization to subdivide the acoustic space into regions, and lists the Gaussians worth evaluating within each region. Applied to unclustered genonic recognition systems, this technique has allowed us to reduce by more than a factor of five the number of Gaussians considered each frame. Here we apply Gaussian shortlists to the clustered system described in Section 4.1. Several methods for generating improved, smaller Gaussian shortlists are discussed and applied to the same system.

Table 7 shows the word error rates for shortlists generated by a variety of methods. Through a series of methods, we have reduced the average number of Gaussian distributions evaluated for each genone from 18 to 2.48 without compromising accuracy. The various shortlists tested were generated in the following ways:

- **None**: No shortlist was used. This is the baseline case from the clustered system described above. All 18 Gaussians are evaluated whenever a genone is active.
- **12D-256**: Our original shortlist method was used. This method uses a cepstral vector quantization codebook (12-dimensions, 256 codewords) to partition the acoustic space. With unclustered systems, this method generally achieves a 5:1 reduction in Gaussian computation. In this clustered system, only a 3:1 reduction was achieved, most likely because the savings from clustering and Gaussian shortlists overlap.
- **39D-256**: The cepstral codebook that partitions the acoustic space in the previous method ignores 27 of the 39 feature dimensions. By using a 39-dimensional, 256-codeword VQ codebook, we created better-differentiated acoustic regions, and reduced the average shortlist length to 4.93.
- **39D-4096**: We further decreased the number of Gaussians per region by shrinking the size of the regions. Here we used a single-feature VQ codebook with 4096 codewords.

For such a large codebook, vector quantization is accelerated using a binary tree VQ fastmatch.

- 39D-4096-min1: When generating a Gaussian shortlist, certain region/genone pairs with low probabilities are assigned very few or even no Gaussians densities. When we were using 48 Gaussians/genone, we found it important to ensure that each list contains a minimum of three Gaussian densities. With our current clustered systems we found that we can achieve similar recognition accuracy by ensuring only one Gaussian per list. As shown in Table 7, this technique results in lists with an average of 2.48 Gaussians per genone, without hurting accuracy.

Shortlist	Shortlist Length	Gaussians Evaluated per Frame	Word Error (%)
none	18	5459	13.64
12D-256	6.08	1964	13.53
39D-256	4.93	1449	13.46
39D-4096	3.68	1088	13.64
39D-4096-min1	2.48	732	13.50

Table 7: Word error rates and Gaussians evaluated, for a variety of Gaussian shortlists

Thus, with the methods in Sections 4.1 and 4.2, we have used clustering, retraining, and new Gaussian shortlist techniques to reduce computation from 48 to an average of 2.48 Gaussians per genone without affecting accuracy.

4.3. Genonic Approximations

We have successfully employed one other method for reducing Gaussian computation. For certain pairs of genones and acoustic regions, even the evaluation of one or two Gaussian distributions may be excessive. These are cases where the output probability is either very low or very uniform across an acoustic region. Here a uniform probability across the region (i.e., requiring no Gaussian evaluations) may be sufficient to model the output probability.

To provide these regional flat probabilities, we implemented a discrete-density HMM, but one whose output probabilities were a region-by-region approximation of the probabilities of our genonic system. Since the two systems' outputs are calibrated, we can use them interchangeably, using a variety of criteria to decide which system's output to use for any given frame, state, acoustic region, or hypothesis. This technique, using variable resolution output models for HMMs is similar to what has been suggested by Alleva et al. [1].

We train this genonic approximation by averaging, for each acoustic region, the output of each genone across a set of observations. The resulting system can be used either by itself or in combination with the continuous system from which it was trained.

Table 8 shows the performance of the discrete approximate genone systems as a function of the number of regions used.

Genonic System	Number of Acoustic Regions	Word Error (%)
Continuous	N/A	13.64
Discrete	256	31.72
Discrete	1024	23.62
Discrete	4096	20.32
Discrete	16384	18.40

Table 8: Accuracy of genonic approximation systems

Even with 16384 acoustic regions, the discrete genonic approximation has an error rate of 18.40%, compared with the baseline continuous system at 13.64%. However, when these discrete systems are used selectively in combination with a continuous genonic system, the results are more encouraging. Our most successful merger combines the 4096-region discrete approximation system (20.32% error) with the 39D-4096-min1 genone system from Table 7 (13.50% error). In combining the two, instead of ensuring that a single Gaussian density was available for all shortlists, the genonic approximation was used for cases where no densities existed. In this way, we were able to eliminate another 25% of the Gaussian computations, reducing our lattice-based computation burden to 564 Gaussians per frame, with a word error of 13.36%.

In summary, we started with a speech recognition system with 28,272 Gaussian distributions that computed 14,538 Gaussian distributions per frame and achieved a 13.43% word error rate running 12.2 times slower than real time on word lattices. Using the techniques described in Section 4, we have reduced the system's computational requirements to 564 Gaussians per frame, resulting in a system with word error of 13.36%, running at 2.0 times real time on our word lattices.

5. MULTIPASS APPROACHES

The techniques for improving the speed of single-pass speech recognition systems can be combined to achieve other speed/accuracy trade-offs (e.g., trees using genone systems with reduced Gaussian computation rates). Furthermore, with multipass approaches [8,9] many of these techniques can be used independently as the different passes of the speech recognition system. For instance, a discrete density tree search may be used in a lattice building or a forward pass, and a Gaussian system may be used in the lattice and/or backward passes.

We have performed preliminary evaluations of several of the tree-based systems presented in Section 3 to evaluate their performance as forward passes for a forward-backward search [9]. Preliminary results show that forward tree-based systems with 30% word error would add at most 3% to the word error rate of a full accuracy backward pass (i.e., at most increase the error rate from

approximately 10% to approximately 13%). More detail on this work will be presented at the HLT conference and will be included in the final version of this paper.

6. CONCLUSIONS

Tree-based techniques, combined with appropriate modeling alternatives, can achieve real-time performance at about 30% error rate for ARPA's 20,000-word Wall Street Journal task. We have shown techniques that reduce the computational complexity of more accurate but slower modeling alternatives so that they are near the speed necessary for real-time performance in a multipass search. Our near-future goal is to combine these two technologies so that real-time, high-accuracy large-vocabulary speech recognition can be achieved.

ACKNOWLEDGMENT

We gratefully acknowledge support for this work from ARPA through Office of Naval Research Contract N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Government funding agencies.

REFERENCES

1. F. Alleva, X. D. Huang and M.-Y. Hwang, "An Improved Search Algorithm Using Incremental Knowledge for Continuous Speech Recognition," *Proc. ICASSP*, pp. II-307 - II-310, April 1993.

2. E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," *Proc. ICASSP*, pp. II-692 - II-695, April 1993.

3. V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," to appear in *Proc. ICASSP*, 1994.

4. V. Digalakis, P. Monaco and H. Murveit, "Acoustic Calibration and Search in SRI's Large Vocabulary HMM-based Speech Recognition System," *Proc. IEEE ASR Workshop*, Snowbird, Dec. 1993.

5. M.-Y. Hwang and X. D. Huang, "Subphonetic Modeling with Markov States - Senone," *Proc. ICASSP*, pp. I-33-36, March 1992.

6. A. Kannan, M. Ostendorf and J. R. Rohlicek, "Maximum Likelihood Clustering of Gaussians for Speech Recognition," in *IEEE Transactions Speech and Audio Processing*, to appear July 1994.

7. H. Ney, R. Haeb-Umbach, B. Tran and M. Oerder, "Improvements in Beam Search for 10,000-word Continuous Speech Recognition," *Proc. ICASSP*, pp. I-9 - I-12, March 1992.

8. H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large Vocabulary Dictation using SRI's DECIPHER™ Speech Recognition System: Progressive Search Techniques," *Proc. ICASSP*, pp. II-319 - II-322, April 1993.

9. L. Nguyen, R. Schwartz, F. Kubala and P. Placeway, "Search Algorithms for Software-Only Real-Time Recognition with Very Large Vocabularies," *Proc. ARPA Human Language Technology Workshop*, March 1993.

10. L. Bahl, S. De Gennaro, P. Gopalakrishnan and R. Mercer, "A Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition," *Proc. Eurospeech 1989*.

11. S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *ASSP-35* pp. 400-401, March 1987.

SRI NOVEMBER 1993 CSR SPOKE EVALUATION

Mitchel Weintraub, Leonardo Neumeyer and Vassilios Digalakis

SRI International
Speech Technology and Research Laboratory
Menlo Park, CA, 94025

ABSTRACT

In this paper we present SRI's results on the 1993 ARPA CSR Spoke Evaluations. This evaluation used the same HMM acoustic models as those used in SRI's hub system: gender-dependent Genonic HMM's. The system was made robust by modifying the front end algorithms to estimate the cepstral features (the HMM models were not modified). The robust front-end used a wide bandwidth (100-6400Hz) and estimated the cepstral coefficients using a series of algorithms that had little effect on the Sennheiser features while making the secondary microphone features look more like the Sennheiser features. The decoder used SRI's DECIPHER™ speech recognition system [1-5] with a progressive search multipass HMM system, and used the Lincoln Lab 5K NVP trigram language model.

1. SYSTEM DESIGN HIGHLIGHTS

An overview of the SRI robust system design used for spokes S5, S6 and S7 is shown below:

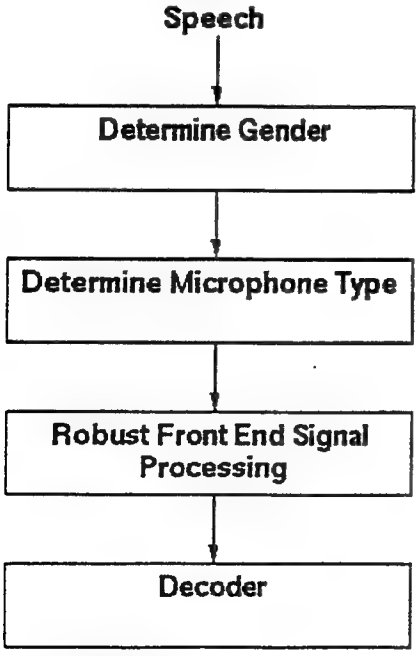

Figure 1: SRI's Robust CSR Block Diagram

2. RESULTS HIGHLIGHTS

SRI's results on Spoke S5 demonstrated that for unknown workstation microphones, there was an overall increase in the word-error rate of 27% over the Sennheiser microphone. For the Audio-Technica Microphone in Spoke S6, there was an 8% increase in word-error rate over the Sennheiser microphone. There was no significant difference between the performance using the Sennheiser microphone and the Audio-Technica Microphone. This is the first time that no significant increase in word-error was observed when training with a high-quality close-talking microphone and testing with a secondary microphone.

SRI's experiments demonstrated that unknown microphone algorithms can outperform known-microphone algorithms. By designing speech-recognition systems that use information about many different microphones, a recognition system can be designed so that small amounts of information about a new environment will not be sufficient to improve performance. The recognition analogy is that speaker-independent system with lots of training can outperform a speaker dependent system with only limited training. For this reason, SRI's system for Spokes S6 and S7 used our best robust-system from Spoke S5. This has led to some confusion for the P0 condition for these spokes. In summary, for Spokes S6 and S7 we did not think that it was necessary to adapt to the new microphone conditions as our microphone-independent system was robust to channel and noise conditions.

3. GENDER/MICROPHONE SELECTION

The gender selection algorithm consisted of a two stage process. The first stage was a fast initial gender decision which used a single state HMM model for each gender (one state for male speech, one state for female speech). Each state used 256 Gaussian mixtures to represent the speech features. The features used for initial gender determination was the baseline zero-mean cepstral features C1-C12 augmented with pitch information.

After the initial gender selection, progressive-search word lattices [3] were generated with speech-recognition models of the initial gender. These word-lattices were used to score the input utterance with a full-HMM system of each gender. The full-HMM models were then used to make the final gender selection based on HMM probability. If the HMM models reversed the decision of the earlier classifier, then new progressive-search word-lattices were generated for this sentence.

The results of gender selection are shown below in the following table:

Experiment	Sennheiser		Secondary Microphone	
	Fast	HMM	Fast	HMM
Spoke S5	0.0%	0.0%	4.2%	2.3%
Spoke S6	0.7%	0.0%	2.5%	0.2%
Spoke S7	2.0%	0.9%	4.9%	2.0%

Table 1: Sentence Misclassification Rate on Eval Test Set

The results in Table 1 show that the HMM classifier has a 65% lower misclassification error rate compared to the fast algorithm. Note that for the conditions when no noise is present, the HMM classifier never makes a sex selection error. When tested in noise, the error rate rises to 0.9%. The average gender classification error rate for the secondary microphone (averaged across all spokes S5-S7) is 1.4%, which is slightly higher than the rate when noise is present for the Sennheiser waveform.

After the gender of the waveform has been determined, the robust system then automatically determines the type of microphone that the speech system was collected with. This is done using a fast single state-HMM classifier, similar to the one described above. The difference here is that only the baseline zero-mean cepstral features C1-C12 are used to determine the microphone type (pitch information is not used). A summary of the overall microphone classification system is shown below in Figure 2.

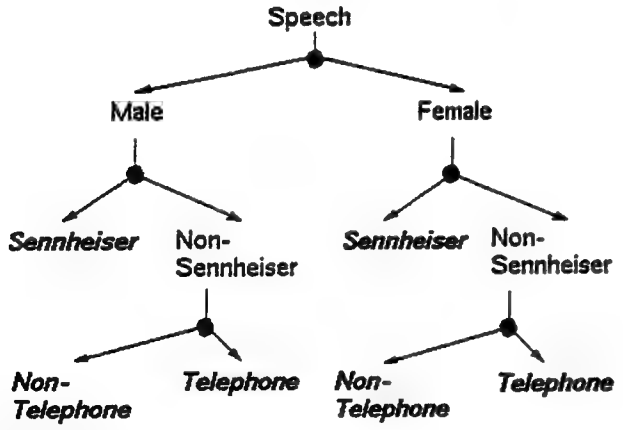

Figure 2: Selection of Microphone Type: (a) Sennheiser, (b) Non-Sennheiser & Non-Telephone, (c) Telephone

The classification rate of the above algorithm is shown in Table 2. Note that in high-noise environments, the Sennheiser waveform becomes misclassified as a secondary microphone almost 30% of the time. When this happens, the robust signal processing is then applied to these Sennheiser waveforms. Also note that the error is never reversed: no secondary microphone waveforms in noise are classified as Sennheiser waveforms. Although SNR is not an explicit feature for determining microphone type, this information is clearly represented in the cepstral features. Also, there are no classification errors in distinguishing between workstation-microphones and telephone-microphone waveforms.

	Sennheiser	Secondary Microphone
Spoke S5	2.3%	8.8%
Spoke S6	1.4%	0.0%
Spoke S7	29.2%	0.0%

Table 2: Classification Errors for Distinguishing Between Sennheiser and Non-Sennheiser Waveforms

4. SELECTION of BEST FRONT END and ROBUST CEPSTRAL COMPUTATION

Each of the different terminal conditions in Figure 2 has a different front end associated with it. For the Sennheiser waveforms, the baseline zero-mean cepstra were used as the features. For both telephone and non-telephone front end systems, different robust estimation algorithms were used to estimate the cepstral features. An overview of the cepstral feature computation is shown below in Figure 3.

- **Single POF Mapping Models**

- **Multiple Parallel Mapping Models**

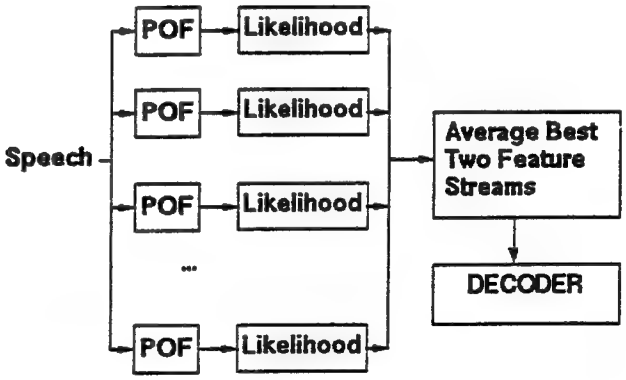

Figure 3: Use of POF Robust Front End for Unknown Microphone: Single or Multiple Front End Mappings.

The above description in Figure 3 illustrates the two algorithms used to estimate robust cepstral features. Fort the non-Sennheiser, non telephone condition, we used the first approach of a single POF mapping. For the telephone condition, we used the second approach of multiple parallel mappings. The results used to select different conditions can be found in Section 7.4.

An overview of the subsequent steps in recognizing with the robust speech recognition system is described below:

- Generate Robust Acoustic Features According To Gender and Microphone Type
- Generate N-Best Lists for Rescoring Using Gender-Dependent Genonic HMM Acoustic Models
  - Progressive Search Uses Initial Word Lattices Generated During Sex Selection Stage
- Rescore N-Best Lists Using Lincoln Lab. Trigram Language Model

The following two sections describe the computation of robust cepstral features. The decoding stages used in the latter stages of processing are described elsewhere [3].

4.1. Non-Telephone Cepstra Computation

A summary of the workstation-microphone cepstral computation is described below:

- Uses Single POF Mapping Model.
- Train Mapping Using 600 Stereo Waveform Pairs from 11 Different Microphones.
  - Used SI-Many Portion of WSJ0 + WSJ1 Corpus
- Compute 256 Mapping Transformations: One For Each of 256 Regions of Acoustic Space.
  - Frame Filterbank Instantaneous SNR
  - 25 Dimensional Feature
  - Current Frame Augmented with Neighboring 2 Frames On Each Side
- Separate Mapping for C0 and Cepstra C1-C12.

The parameters that the above algorithm refers to are described in more detail in section 6 and 7 which describe the POF model.

4.2. Telephone Cepstra Computation

A summary of the telephone-microphone cepstral computation is described below:

- Use 15 Parallel Mapping Models.
  - One Mapping Model for Each of 14 Microphones
  - Baseline Zero-Mean Cepstra is 15th Model
- Train Each Mapping Model Using 200 Stereo Waveform Pairs.
  - Used SI-Many Portion of WSJ0 + WSJ1 Corpus
- Compute Features Using Each Mapping.
- Select Best Two Feature Streams.
  - Compute Likelihood of Each Set of Feature for Sentence
  - Average Two Transformed Feature Streams: C0-C12
  - Compute First and Second Derivatives on Averaged Features.

Section 7.4 describes the experimental results in more detail that were used to select the appropriate front end signal processing for this condition.

5. MAPPING ALGORITHMS

5.1. Background

In many practical situations an automatic speech recognizer has to operate in several different but well-defined acoustic environments. For example, the same recognition task may be implemented using different microphones or transmission channels. In this situation it may not be practical to recollect a speech corpus to train the acoustic models of the recognizer. To alleviate this problem, we propose an algorithm that maps speech features between two acoustic spaces. The models of the mapping algorithm are trained using a small database recorded simultaneously in both environments.

In the case of steady-state additive homogenous noise, we can derive a MMSE estimate of the clean speech filterbank-log energy features using a model for how the features change in the presence of this noise [6-7]. In these algorithms, the estimated speech spectrum is a function of the global spectral SNR, the instantaneous spectral SNR, and the overall spectral shape of the speech signal. However, after studying simultaneous recordings made with two microphones, we believe that the relationship between the two simultaneous features is nonlinear. We therefore propose to use a piecewise-nonlinear model to relate the two feature spaces.

5.2. Related Work on Feature Mapping

There have been several algorithms in the literature which have focused on experimentally training a mapping between the noisy features and the clean features [8-13]. This work builds on similar robust algorithm development that have been developed at BBN, CMU, and IBM. Several of the features of these systems have been incorporated into the design of SRI's robust front end analysis.

The proposed algorithm differs from previous algorithms in several ways:

- The MMSE estimate of the clean speech features in noise is trained experimentally rather than with a model as in [6, 7].
- Several frames are joined together similar to [13].
- The conditional PDF is based on a generic noisy feature not necessarily related to the feature that we are trying to estimate. For example, we could condition the estimate of the cepstral energy on the instantaneous spectral SNR vector.
- Multidimensional least-squares filters are used for the mapping transformation. This is used to exploit the correlation of the features over time and among components of the spectral features at the same time.
- Linear transformations are combined together without hard decisions.

- All delta parameters are computed after mapping the cepstrum and cepstral energy.
- The mapping parameters are trained using stereo recordings with two different microphones. Once trained, the mapping parameters are fixed.
- The mapping can be used to map either noisy speech features to clean features during training, or clean features to noisy features during recognition.

5.3. Related Work on Adaptation

The algorithm used to map the incoming features into a more robust representation has some similarities to work on model adaptation. Some of the high-level differences between HMM model adaptation and the mapping algorithms proposed in this paper are:

- The mapping algorithm works by primarily correcting shifts in the mean of the feature set that are correlated with observable information. Adapting HMM model parameters has certain degrees of freedom which the mapping algorithm does not have: e.g. ability to change state variances, and mixture weights.
- Two HMM states that have identical probability distributions which and are not tied can have different distributions after adaptation. These distributions cannot be differentiated by mapping features.
- The mapping algorithms described in this paper are able to incorporate many pieces of information that have been traditionally difficult to incorporate into HMM models and into adaptation algorithms. These include observations which span across several frames and the correlation of the state features with global characteristics of the speech waveform.

These two techniques are not mutually exclusive and can be used together to achieve robust speech recognition performance. The boundary between these two techniques can be blurred when the mapping algorithm is dependent on the speech recognizer's hypothesis.

6. THE POF ALGORITHM

The mapping algorithm is based on a probabilistic piecewise-nonlinear transformation of the acoustic space that we call *Probabilistic Optimum Filtering* (POF). Let us assume that the recognizer is trained with data recorded with a high-quality close-talking microphone (clean speech), and the test data is acquired in a different acoustic environment (noisy speech). Our goal is to estimate a clean feature vector $\hat{\mathbf{x}}_n$ given its corresponding noisy feature y_n where n is the frame index. (A list of symbols is shown in Table 1.) To estimate the clean vector we vector-quantize the clean feature space in I regions using the generalized Lloyd algorithm [14]. Each VQ region is assigned a multidimensional transversal filter (see Figure 1). The error between the clean vector and the estimated vectors produced by the i-th filter is given by

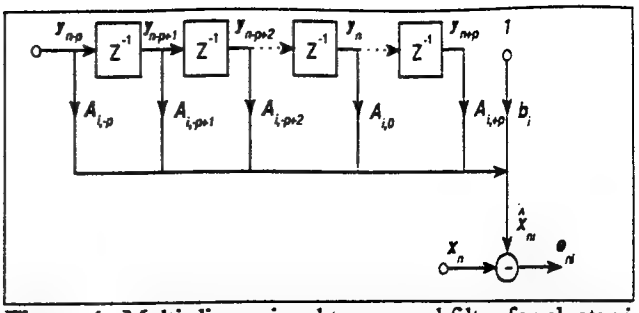

Figure 4: Multi-dimensional transversal filter for cluster i.

$$e_{ni} = x_n - \hat{x}_{ni} = x_n - W_i^T Y_n \tag{1}$$

where e_{ni} is the error associated with region i, W_i is the filter coefficient matrix, and Y_n is the tapped-delay line of the noisy vectors. Expanding these matrices we get

$$W_i^T = \left[A_{i,-p} \cdots A_{i,-1}\, A_{i,0}\, A_{i,1} \cdots A_{i,p}\, b_i\right] \tag{2}$$

$$Y_n^T = \left[y_{n-p}^T \cdots y_{n-1}^T\, y_n^T\, y_{n+1}^T \cdots y_{n+p}^T\, 1\right] \tag{3}$$

The conditional error in each region is defined as

$$E_i = \sum_{n=p}^{N-1-p} \|e_{ni}\|^2 p(g_i|z_n) \tag{4}$$

where $p(g_i|z_n)$ is the probability that the clean vector x_i belongs to region g_i given an arbitrary conditional noisy feature vector z_n. Note that the conditioning noisy feature can be any acoustic vector generated from the noisy speech frame. For example, it may include an estimate of the signal-to-noise ratio (SNR), energy, cepstral energy, cepstrum, etc.

The conditional probability density function $p(z_n|g_i)$ is modeled as a mixture of I Gaussian distributions. Each Gaussian distribution models a VQ region. The parameters of the distributions (mean vectors and covariance matrices) are estimated using the corresponding z_n vectors associated with that region. The posterior probabilities $p(g_i|z_n)$, are computed using Bayes' theorem and the mixture weights, $p(g_i)$, are estimated using the relative number of training clean vectors that are assigned to a given VQ region.

To compute the optimum filters in the mean-squared error sense, we minimize the conditional error in each VQ region. The minimum mean-squared error vector is obtained by taking the gradient of E_i defined in Eq. (4) with respect to the filter coefficient matrix and equating all the elements of the gradient matrix to zero. As a result, the optimum filter coefficient matrix has the form, $W_i = R_i^{-1} r_i$ where

Symbol	Dimension	Description
n	1	frame index
i	1	region index
L	1	feature vector size
M	1	conditioning feature vector size
N	1	number of training frames
I	1	number of VQ regions
p	1	maximum filter delay
e_{ni}	$L \times 1$	estimation error vector
x_n	$L \times 1$	clean feature vector
$\hat{x}_n$	$L \times 1$	estimate of clean feature vector
y_n	$L \times 1$	noisy feature vector
z_n	$M \times 1$	conditioning noisy feature vector
μ_i	$M \times 1$	mean vector of gaussian i
Σ_i	$M \times M$	covariance matrix of gaussian i
W_i	$(2p+1)L+1 \times L$	transversal filter coefficient matrix
Y_n	$(2p+1)L+1 \times 1$	tap input vector
A_{ik}	$L \times L$	multiplicative tap matrix
b_i	$L \times 1$	additive tap matrix
R_i	$(2p+1)L+1 \times (2p+1)L+1$	auto-correlation matrix
r_i	$(2p+1)L+1 \times L$	cross-correlation matrix

Table 3: List of Symbols.

$$R_i = \sum_{n=p}^{N-1-p} Y_n Y_n^T p(g_i|z_n) \tag{5}$$

is a probabilistic non-singular auto-correlation matrix, and

$$r_i = \sum_{n=p}^{N-1-p} Y_n x_n^T p(g_i|z_n) \tag{6}$$

is a probabilistic cross-correlation matrix.

The algorithm can be completely trained without supervision and requires no additional information other than the simultaneous waveforms.

The run-time estimate of the clean feature vector can be computed by integrating the outputs of all the filters as follows:

$$\hat{x}_n = \sum_{i=0}^{I-1} W_i^T Y_n p(g_i|z_n) = \left\{ \sum_{i=0}^{I-1} W_i^T p(g_i|z_n) \right\} Y_n \tag{7}$$

7. EXPERIMENTS

7.1. Introduction

In this section we present a series of experiments that show how the mapping algorithm can be used in a continuous speech recognizer across acoustic environments. In all of the experiments the recognizer models are trained with data recorded with high-quality microphones and digitally sampled at 16,000 Hz. The analysis frame rate is 100 Hz.

The tables below show three types of performance indicators:

- *Relative distortion measure*. For a given component of a feature vector we define the relative distortion between the clean and noisy data as follows:

$$d = \sqrt{\frac{E\left[(x-y)^2\right]}{\mathrm{var}(x)}} \tag{8}$$

- *Word recognition error.*
- *Error ratio*. The error ratio is given by E_n/E_c where E_n is the word recognition error for the test-noisy/train-clean condition, and E_c is the word recognition error of the test-clean/train-clean condition.

7.2. Single Microphone

To test the POF algorithm on a single target acoustic environment we used the DARPA *Wall Street Journal* database [15] on SRI's DECIPHER™ phonetically tied-mixture speech recognition system [2]. The signal processing consisted of a filterbank-based front-end that generated six feature streams: cepstrum *(c1-c12)*, cepstral energy *(c0)*, and their first- and second-order derivatives. Cepstral-mean normalization [16] was used to equalize the channel. We used simultaneous recordings of high-quality speech (Sennheiser 414 head-mounted microphone with a noise-cancelling element) along with speech recorded by a standard speaker phone (AT&T 720) and transmitted over local telephone lines. We will refer to this stereo data as *clean* and *noisy* speech respectively. The models of the recognizer were trained using 42 male WSJ0 training talkers (3500 sentences) recorded with a Sennheiser microphone. The models of the mapping algorithm were trained using 240 development training sentences recorded by three speakers. The test set consisted of 100 sentences (not included in the training set) recorded by the same three speakers.

In this experiment we mapped two of the six features: the cepstrum *(c1-c12)* and the cepstral energy *(c0)* separately. The derivatives were computed from the mapped vectors of the cepstral features. For the conditioning feature we used a 13-dimensional cepstral vector *(c0-c12)* modeled with 512 Gaussians with diagonal covariance matrices. The results are shown in Table 2.

Filter Coefficients	Average Distortion	Recognition Error (%)	Error Ratio
No mapping	0.72	27.6	2.46
$A_{i,0}$=I, b_i	0.62	18.1	1.62
$A_{i,0}$, b_i	0.57	17.0	1.52

Table 4: Performance of the POF algorithm for different number of filter coefficients. The number of Gaussian distributions is 512 per feature and the conditioning feature is a 13-dimensional cepstral vector.

Filter Coefficients	Average Distortion	Recognition Error (%)	Error Ratio
$A_{i,-1}$ ..., $A_{i,-1}$, b_i	0.51	17.3	1.54
$A_{i,-2}$ ..., $A_{i,-2}$, b_i	0.50	16.4	1.46
$A_{i,-3}$ ..., $A_{i,-3}$, b_i	0.49	15.9	1.42
$A_{i,-4}$ ..., $A_{i,-4}$, b_i	0.49	16.1	1.44

Table 4: Performance of the POF algorithm for different number of filter coefficients. The number of Gaussian distributions is 512 per feature and the conditioning feature is a 13-dimensional cepstral vector.

The baseline experiment produced a word error rate of 27.6% on the noisy test set, that is, 2.46 times the error obtained when using the clean data channel. A 34% improvement in recognition performance was obtained when using only the additive filter coefficient b_i. (Recognition error goes down to 18.1%.) The best result (15.9% recognition error) was obtained for the condition $p=3$, in which six neighboring noisy frames are being used to estimate the feature vector for the current frame. The correlation between the average relative distortion between the six clean and noisy features and the recognition error is 0.9.

7.3. ATIS Simultaneous Corpus

To test the performance of the POF algorithm on multiple microphones we used SRI's stereo-ATIS database. (See [1] for details.) A corpus of both training and testing speech was collected using simultaneous recordings made from subjects wearing a Sennheiser HMD 414 microphone and holding a telephone handset. The speech from the telephone handset was transmitted over local telephone lines during data collection. Ten different telephone handsets were used. Ten talkers were designated as training talkers, and three talkers were designated as the test set. The training set consisted of 3,000 simultaneous recordings of Sennheiser microphone and telephone speech. The test set consisted of 400 simultaneous recordings of Sennheiser and telephone speech. The results obtained with this pilot corpus are shown in Table 5.

Acoustic Model Training		Test Set Word Error (%)	
Training Data	Front End Bandwidth	Sennheiser	Telephone
Sennheiser	Wide	7.8	19.4
Sennheiser	Telephone	9.0	9.7
Telephone	Telephone	10.0	10.3

Table 5: Effect of Different Training and Front-End Bandwidth on Test Set Performance. Results are Word Error Rate on the 400 Sentence Simultaneous Test Set

We can see from Table 5 that there is a 15.4% decrease in performance when using a telephone front end (7.8% increases to 9.0% word error) and testing on Sennheiser data. This is due to the loss of information in reducing the bandwidth from 100-6400 Hz to 300-3300 Hz. However, when we are using a telephone front end, there is only a 7.8% increase in word error when testing on telephone speech compared to testing on Sennheiser speech (9.7% versus 9.0%). This is very surprising result, and we had expected a much bigger performance difference when Sennheiser models are tested on telephone speech acoustics.

7.4. Multiple Microphones: Single or Multiple Mapping

There are a number of ways that the POF mapping algorithm can be used when the microphone is unknown. Some of these variations are shown in Table 6.

Experiment		Word Error
Single Mapping Combining All 10 Telephones in Training Data		9.4
Train 10 Mappings, One for Each Telephone; Run 10 Recognizers in Parallel, each using Different Mapping; Select Recognizer with Highest Probability		9.2
Train 10 Mappings, One for Each Telephone; Run 10 Mappings in Parallel and Average Features of Best N Feature-Streams that Have Highest Likelihood	Top1	9.3
	Top2	9.2
	Top3	8.9
	Top4	8.7
Train 15 Mappings for WSJ Corpus; Run 15 Mappings in Parallel and Average Features of Best N Feature-Streams that Have Highest Likelihood	Top1	9.8
	Top2	9.6
	Top3	10.3
	Top4	10.7

Table 6: Performance on the multiple-telephone handset test set when mapping algorithm is used in different ways.

The differences between the experimental conditions are small, but the trends are different and depend on the mapping and the corpus. These differences depend on the similarities of the different microphones that are used in training conditions, and the relationship between the training and the testing conditions.

When the microphones are all similar (10 telephone mappings), then averaging the features of each mapping helps improve performance. When the microphones are very different (e.g. those in WSJ corpus), averaging the features of each mapping has a minimum when averaging two best (likelihood) feature streams.

7.5. Multiple Microphones: Conditioning Feature

The next experiment varied the conditioning feature. The conditioning feature is the feature vector that is used to divide the space into different acoustic regions. In each region of the acoustic space there is a different linear transformation that is trained.

The mapping approach was fixed: we used a single POF mapping for multiple telephone handsets. For this experiment we mapped the cepstrum vector *(c1-c12)* and the cepstral energy *(c0)*. The maximum delay of the filters was kept fixed at $p=2$, and the number of Gaussians was 512. The experimental variable was what feature the estimates were conditioned on. We tried the following conditioning features:

- **Cepstrum.** Same conditioning feature used in the single microphone experiment *(c0-c12)*.
- **Spectral SNR.** This is an estimate of the instantaneous signal-to-noise ratio computed on the log-filterbank energy domain. The vector size is 25.
- **Cepstral SNR.** This feature is generated by applying the discrete cosine transform (DCT) to the spectral SNR. The transformation reduces the dimensionality of the vector from 25 to 12 elements.

The results are shown in Table 7. The baseline result is a 19.4% word error rate. This result is achieved when the same wide-band front-end is used for training the models with clean data and for recognition using telephone data. When a telephone front-end [1] is used for training and testing, the error decreases to 9.7%. The disadvantage of using this approach is that the acoustic models of the recognizer have to be re-estimated. However, the POF-based front-end operates on the clean models and results in better performance. The cepstral SNR produces the best result (8.7%). With this conditioning feature we combine the effects of noise and spectral shape in a compact representation.

Experiment	Word Error (%)	Error Ratio
Wide-band front-end	19.4	2.49
Telephone-bandwidth front-end	9.7	1.24
Mapping with cepstrum	9.4	1.20
Mapping with spectral SNR	8.9	1.14
Mapping with cepstral SNR	8.7	1.11

Table 7: Performance for the multiple-telephone handset test set when varying the conditioning feature.

8. WSJ EXPERIMENTAL RESULTS

Another series of experiments was performed on the *Wall Street Journal* (WSJ) Speech Corpus [15]. We evaluated our system on the 5000-word-recognition closed-vocabulary speaker-independent speech-recognition tasks: Spoke S5 Unknown Microphone, Spoke S6: known microphone, and Spoke S7 Noisy Environment.

The version of the DECIPHER speaker-independent continuous speech recognition system used for these experiments is based on a progressive-search strategy [3] and continuous-density, GENONIC hidden Markov models (HMMs) [2]. Gender-dependent models are used in all passes. Gender selection is accomplished by selecting the gender with the higher recognition likelihood.

The acoustic models used by the HMM system were trained with 37,000 sentences of Sennheiser data from 280 speakers, a set officially designated as the WSJ0+WSJ1 many-speaker baseline training. A 5K closed-vocabulary back-off trigram language model provided by M.I.T. Lincoln Laboratory for the WSJ task was used. Gender-dependent HMM acoustic models were used.

The front-end processing extracts one long spectral vector which consists of the following six feature components: cepstrum, energy and their first and second order derivatives. The dimensionality of this feature is 39 (13 * 3) for the wide-bandwidth spectral analysis and 27 (9 * 3) for the telephone-bandwidth spectral analysis. The cepstral features are computed from an FFT filterbank, and subsequent cepstral-mean normalization on a sentence by sentence basis is performed.

Before using wide-bandwidth context-dependent genonic HMMs, a robust estimate of the Sennheiser cepstral parameters is computed using Probabilistic Optimum Filtering. The robust front-end analysis is designed for an unknown microphone condition. The POF mapping algorithm estimates are conditioned on the noisy cepstral observations. Separate mappings are trained for each of the 14 microphones in the baseline WSJ0+WSJ1 si_tr_s stereo training, as well as one mapping for the overall case of single non-telephone mapping. When the default no-transformation zero-mean cepstra are included, this makes a total of 15 estimated feature streams. These feature streams are computed on each test waveform, and the two feature streams with the highest likelihoods (using a simplified HMM for scoring the features) are averaged together (Top2). In all cases the first and second delta parameters are computed on these estimated cepstral values.

Front-End Bandwidth	Signal Processing	Test Set	Word Error (%)
Wide	Standard	Sennheiser	5.8
Telephone	Standard	Sennheiser	9.6
Telephone	Standard	Telephone	10.9
Wide	Robust POF15 Cepstral Mapping	Telephone	11.9

Table 8: Performance on the Aug 1993 WSJ Spoke S6 Development Test Set for Simultaneous Sennheiser/Telephone Recordings

The results in Table 8show that most of the loss in performance between recognizing on high-quality Sennheiser recordings and on local telephone speech is due to the loss of information outside the telephone bandwidth. There is an increase in the word-error rate of 66% when testing on Sennheiser recordings with a wide-bandwidth analysis (5.8%) compared to testing with a telephone-bandwidth analysis (9.6%).

The loss in performance when switching from Sennheiser recordings to telephone recordings is small in comparison to the loss of information due to bandwidth restrictions. There is a 14% increase in the word-error rate when testing on the Sennheiser recordings (9.6%) compared to testing on the AT&T telephone recordings (10.9%).

8.1. Official Spoke Results: Unknown Microphone

The results in Table 9show the speech recognition performance when the secondary microphone condition is unknown. In these experiments, the robust signal processing front end decreased the word error rate from 17.2 to 13.1%.

Experiment	Word Error	
	Sennheiser	Secondary Microphone
Compensation Disabled	6.6	17.2
Compensation Enabled	6.6	13.1

Table 9: Word Error Rate With and Without Compensation on both Sennheiser and Secondary Microphone Data

If we look in more detail at the results, we see the following information:

Spoke S5 Condition	P0	C1	C2	C3	P0/C3
2 Telephones	29.9	48.7	4.2	4.2	7.12
8 Non-Telephones	9.3	10.2	7.3	7.3	1.27

Table 10: Spoke S5 Results for Telephone and Non-Telephone Microphones.

Clearly, the robust algorithms did not perform well for telephone speech. We attribute this to the fact that we used wide bandwidth acoustic models and the robust front end was not able to accurately estimate the cepstral features for this test data. The poor official telephone results are due to out-of-band noise which was correlated with the speech signal that our system was sensitive to. This can also be seen in the poor official telephone results for spokes S6 and S7 as well. In retrospect, we should have used telephone bandwidth HMM models for this condition.

Note that for the workstation microphones, there was an increase of only 27% in the word-error over the Sennheiser microphone condition.

8.2. Official Spoke Results: Known Microphone

The results in Table 11 show no significant difference in speech recognition performance between those obtained with the Audio-Technica microphone and those obtained with the Sennheiser microphone. The robust front-end signal processing has demonstrated for the first time that one can achieve the same performance with a stand-mounted microphone as with a high-quality close-talking microphone, all when trained on high-quality speech corpus.

Experiment	Word Error	
	Sennheiser	Secondary Microphone
Audio-Technica Recordings	5.9	6.4
Telephone Handset Recordings	7.2	19.1

Table 11: Word Error for both Sennheiser and Secondary Microphone with Robust Signal Processing Front End

Note that the above system did not use any microphone adaptation data. For this reason, we did not have a P0 result. It was our belief that since there was no degradation in performance when changing microphone, that no microphone adaptation algorithms were needed. In particular, the results from our development testing are shown below:

Test Microphone	Signal Processing	Word Error	Error Ratio
Sennheiser	Standard	7.7	1.00
Audio-Technica	Standard	10.9	1.42
Audio-Technica	POF W/Mic Adapt	9.2	1.19
Audio-Technica	ROBUST POF15	8.7	1.13

Table 12: Results on 4 Male Talkers in Spoke S6 Audio-Technica Development Test Set Using a Pre-Evaluation Recognizer With Bigram Grammar

Note that the best robust system had an error ratio (secondary microphone error-rate over Sennheiser-microphone error rate) which increased the error by only 13% over the Sennheiser condition, while the microphone adaptation system had an error increase of 19%. Both systems were better than the baseline cepstral zero-mean system which had an increase in the error rate of 42%.

8.3. Official Spoke Results: Noisy Environment

The results in Table 13 show the performance when the recordings are made in a noisy environment. The first noisy environment was a computer room (average background noise level of 58-59 dBA), and the second noisy environment was a laboratory with mail sorting equipment (average noise level varied from 62-68 dBA). The error rates are significantly higher for the audio-technica microphone than the sennheiser microphone in the noisier environment. In the computer room environment, the performance with the audio-technica microphone is almost indistinguishable from that of the Sennheiser recording.

Experiment		Word Error	
		Sennheiser	Secondary Microphone
Audio-Technica Recordings	Env 1	6.3	8.5
	Env 2	9.1	17.4
Telephone Handset Recordings	Env 1	8.4	29.1
	Env 2	8.3	28.8

Table 13: Word Error for both Sennheiser and Secondary Microphone with Robust Signal Processing Front End when Recorded in Two Noisy Environments

For the Audio-Technica recordings, there was a 35% increase in the word-error rate when used in a computer-room environment, but a 90% increase in the word-error rate when used in the mail-sorting equipment environment.

9. CONCLUSIONS

We have presented a feature mapping algorithm capable of exploiting nonlinear relations between two acoustic spaces. We have shown how to improve the performance of the recognizer in the presence of a noisy signal by using a small database with simultaneous recordings in the clean and noisy acoustic environments.

We have shown that:

- There is no significant difference in speech recognition performance between those obtained with the Audio-Technica microphone and those obtained with the Sennheiser microphone. There is no significant performance degradation in a quiet environment and only a slight degradation in low noise environments (~59 dBA).
- Multidimensional least-squares filters can be successfully used to exploit the correlation of the features over time and among components of the spectral features at the same time. These filters can be conditioned on both local & global spectral information to improve robust recognition performance.
- Most of the performance loss in converting wide-bandwidth models to telephone speech models is due to the loss of information associated with the telephone bandwidth.
- It is possible to construct acoustic models for telephone speech using a high-quality speech corpus with only a minor increase in recognition word-error rate.
- A telephone-bandwidth system trained with high-quality speech can outperform a system that is trained on telephone speech even when tested on telephone speech.
- The variability introduced by the telephone handset does not degrade speech recognition performance.
- Robust signal processing can be designed to maintain speech recognition performance using wide-bandwidth HMM models with a telephone-bandwidth test set.

ACKNOWLEDGMENTS

The authors thank John Butzberger for helping to set up the ATIS experiments.

This research was supported by a grant, NSF IRI-9014829, from the National Science Foundation (NSF). It was also supported by the Advanced Research Projects Agency (ARPA) under Contracts ONR N00014-93-C-0142 and ONR N00014-92-C-0154.

REFERENCES

1. M. Weintraub and L. Neumeyer, "Constructing Telephone Acoustic Models from a High-Quality Speech Corpus," 1994 IEEE ICASSP.
2. V. Digalakis and H. Murveit, "GENONES: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," 1994 IEEE ICASSP.
3. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques," 1993 IEEE ICASSP, pp. II319-II322.
4. H. Murveit, J. Butzberger, and M. Weintraub, "Performance of SRI's DECIPHER™ Speech Recognition System on DARPA's CSR Task," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp. 410-414.
5. Murveit, H., J. Butzberger, and M. Weintraub, "Reduced Channel Dependence for Speech Recognition," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp. 280-284.
6. A. Erell and M. Weintraub, "Spectral Estimation for Noise Robust Speech Recognition," 1989 DARPA SLS Workshop, pp. 319-324.
7. A. Erell and M. Weintraub, "Recognition of Noisy Speech: Using Minimum-Mean Log-Spectral Distance Estimation," 1990 DARPA SLS Workshop, pp. 341-345.
8. B.H. Juang and L.R. Rabiner, "Signal Restoration by Spectral Mapping," 1987 IEEE ICASSP, pp. 2368-2371.
9. H. Gish, Y.L. Chow, and J.R. Rohlicek, "Probabilistic Vector Mapping of Noisy Speech Parameters for HMM Word Spotting," 1990 IEEE ICASSP, pp. 117-120.
10. K. Ng, H. Gish, and J.R. Rohlicek, "Robust Mapping of Noisy Speech Parameters for HMM Word Spotting," 1992 IEEE ICASSP, pp. II-109-II-112.
11. A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition," Ph.D. Thesis, Carnegie-Mellon University, September 1990.
12. R.M. Stern, F.H. Leu, Y. Ohshima, T.M. Sullivan, and A. Acero, "Multiple Approaches to Robust Speech Recognition," 1992 International Conference on Spoken Language Processing, pp. 695-698.
13. A. Nadas, D. Nahamoo, and M. Picheny, "Adaptive Labeling: Normalization of Speech by Adaptive Transformations based on Vector Quantization," 1988 IEEE ICASSP, pp. 521-524.
14. Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Comm., vol. 28, pp. 84-95, January 1980.
15. G. Doddington, "CSR Corpus Development," 1992 DARPA SLS Workshop, pp. 363-366.
16. S.F. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," IEEE Trans. ASSP, Vol. 29, pp. 254-272, April 1981.

Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures*

V. Digalakis D. Rtischev L. Neumeyer

SRI International

333 Ravenswood Ave.

Menlo Park, CA 94025

April 6, 1994

EDICS SA 1.6.7

Abstract

A recent trend in automatic speech recognition systems is the use of continuous mixture-density hidden Markov models (HMMs). Despite the good recognition performance that these systems achieve on average in large vocabulary applications, there is a large variability in performance across speakers. Performance degrades dramatically when the user is radically different from the training population. A popular technique that can improve the performance and robustness of a speech recognition system is adapting speech models to the speaker, and more generally to the channel and the task. In continuous mixture-density HMMs the number of component densities is typically

*Submitted to the IEEE Transactions on Speech and Audio Processing.

very large, and it may not be feasible to acquire a sufficient amount of adaptation data for robust maximum-likelihood estimates. To solve this problem, we propose a constrained estimation technique for Gaussian mixture densities. The algorithm is evaluated on the large-vocabulary Wall Street Journal corpus for both native and nonnative speakers of American English. For nonnative speakers, the recognition error rate is approximately halved with only a small amount of adaptation data, and it approaches the speaker-independent accuracy achieved for native speakers. For native speakers, the recognition performance after adaptation improves to the accuracy of speaker-dependent systems that use 6 times as much training data.

1 Introduction

Recognition error rates ranging from 10% to 15% have recently been achieved in the 20,000-word, open-vocabulary recognition task on the Wall Street Journal (WSJ) corpus [20] using hidden Markov models (HMMs) [2, 12] with continuous-mixture observation densities [19]. However, this recognition performance is far from satisfactory for most usable large-vocabulary recognition (LVR) applications. Moreover, recognition accuracy is very sensitive to speaker variability and will degrade much more in the move from the lab to the field. Speaker-, channel-, or other task-dependent solutions require excessive collection of training data and decrease system utility and portability. A popular technique that can be used to improve the performance and robustness of a speech recognition system is adapting the speech model to the speaker, channel, and task [5, 23, 9, 15]. In this work, we consider adaptation to the speaker, although the techniques can be modified to be used at other levels.

In this paper we will present novel adaptation techniques for state-of-the-art continuous mixture-density HMMs. It has recently been shown that HMMs that use continuous-density probability distributions achieve better recognition performance than those that use discrete-density distributions [19]. After [8], we refer to a group of Gaussians that are used to form a Gaussian mixture distribution as a *genone*, to the collection of these groups as genones, and to HMM systems with an arbitrary degree of genone sharing[1] as genonic HMMs. The degree of genone sharing significantly affects recognition performance [8]. HMM systems with less sharing have typically a smaller number of Gaussians per genone and a larger total number of Gaussians than systems with fewer genones. The increase in the number of

[1] By *degree of genone sharing* we refer to the average number of distinct HMM states that share the same genone's Gaussians in their output distributions.

Gaussians is usually over-compensated for by the decrease in the number of mixture weights, and systems with less sharing have a smaller number of parameters. Hence, they are more suited to adaptation than tied-mixture HMMs (single-genone systems, with all HMM states sharing the same Gaussians in their mixture distributions).

Two families of adaptation schemes have been proposed in the past. One transforms the speaker's feature space to "match" the space of the training population [6, 18, 4]. The transformation can be applied either directly to the features, or to the speech models. The second main family of adaptation algorithms follows a Bayesian approach, where the speaker-independent information is encapsulated in the prior distributions [5, 15]. The transformation approach has the advantage of simplicity. In addition, if the number of free parameters is small, then transformation techniques adapt to the user with only a small amount of adaptation speech (quick adaptation). The Bayesian approach usually has nice asymptotic properties, that is, speaker-adaptive performance will converge to speaker-dependent performance as the amount of adaptation speech increases. However, the adaptation rate is usually slow.

For HMMs with a small degree of sharing and a large total number of Gaussians, it is impractical to expect enough adaptation data to obtain robust maximum-likelihood (ML) estimates of all the Gaussians. To deal with the problem of adapting a large number of Gaussians from small amounts of adaptation speech, we present a new algorithm for the constrained estimation of genones. The algorithm can also be viewed as estimating a transformation of the speaker-independent models by maximizing the likelihood of the adaptation data. In contrast to previous adaptation schemes based on feature transformations, our algorithm has the desirable property of being text-independent. It does not require the new speaker to record sentences with previously specified transcriptions, nor does it require a time warping between the new speaker's utterances and those uttered by the

reference speakers. In Bayesian adaptation techniques, the limited amount of speaker-specific data is combined with the speaker-independent models in an optimal manner. Maximum *a posteriori* (MAP) reestimation for continuous Gaussian-mixture HMMs is equivalent to linearly combining the speaker-dependent sufficient statistics with the speaker-independent priors [16]. Typically, only the Gaussians of the speaker-independent models that are most likely to have generated some of the adaptation data will be adapted to the speaker. This behavior may be problematic for continuous HMMs with a large number of Gaussians, since only a small percentage of the Gaussians will be "seen" in the adaptation data. In contrast, our adaptation scheme can adapt a Gaussian without requiring training examples of this specific Gaussian to exist in the adaptation data. By using a constrained reestimation method, our algorithm is able to extrapolate and adapt Gaussians in a genone based on data that were most likely generated by other Gaussians of the same or other neighboring genones.

This paper is organized as follows. Section 2 presents an algorithm for the constrained estimation of Gaussian mixtures based on the Expectation-Maximization (EM) algorithm. We give the solution for both the static case of a single random vector modeled by a Gaussian mixture density and the dynamic case of a vector process modeled using HMMs with Gaussian mixtures as output distributions. In Section 3 we discuss the application of the main algorithm to the speaker adaptation problem. Section 4 describes experiments and presents results on the WSJ corpus. Finally, discussion of results and conclusions appear in Section 5.

2 Constrained Estimation of Gaussian Mixtures

One speaker adaptation paradigm that fits well with the overall approach of continuous-density HMMs with shared Gaussian codebooks is to employ a transformation of the speaker-independent models to best correspond to the available adaptation data. Such a transformation can be efficiently achieved by assuming that the Gaussians in each genone of the speaker-adapted system are obtained through a transformation of the corresponding speaker-independent Gaussians. This transformation can be either unique for each genone, or shared by different genones. We choose to apply the transformation at the distribution level, rather than transforming the data directly, since we can then use the EM algorithm to estimate the transformation parameters by maximizing the likelihood of the adaptation data. The advantage of using the EM algorithm is that we can estimate the transformation from new-speaker data alone. This eliminates the need of some form of time alignment between the new-speaker data and the training- or reference-speaker data that previous transformation-based techniques needed [6, 18]. The estimation of the transformation can also be viewed as a constrained estimation of Gaussian mixtures.

2.1 Estimation of a Single Gaussian-Mixture

To better illustrate the constrained Gaussian estimation method, we first present the estimation formulae for a single Gaussian-mixture density. In Section 2.2 we extend the method for mixture densities as observation distributions in hidden Markov models. Let us consider a Gaussian mixture density of the form

$$f(x;\theta) = f(x;A,b) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i) N(x; Am_i + b, AS_iA^T), \tag{1}$$

where the model parameters are $\theta = [A, b]$, N_ω is the number of mixture components, and we have the constraint that

$$\sum_{i=1}^{N_\omega} \mathrm{P}(\omega_i) = 1. \tag{2}$$

We assume that the parameters $[m_i, S_i, i = 1, \ldots, N_\omega]$ are fixed, and that the matrices S_i are positive definite.

This model is equivalent to assuming that the random vector x is obtained through an affine transformation $x = Ay + b$ from the unobserved vector y that has a known mixture density

$$g(y) = \sum_{i=1}^{N_\omega} \mathrm{P}(\omega_i) N(y; m_i, S_i). \tag{3}$$

ML estimation of the constrained Gaussian-mixture model is, therefore, equivalent to estimating the regression parameters A, b using only observations of the dependent variable and the knowledge of the distribution of the unobserved variable y.

As shown in [21], the EM algorithm can be used to obtain ML estimates of the parameters of a Gaussian-mixture density in the unconstrained case. The EM algorithm can also be used to estimate the model parameters $[A, b]$ in the constrained case. At each EM iteration, the new parameter estimates are obtained by maximizing the auxiliary function [7]

$$\theta_n = \arg\max_\theta E\{\log \mathrm{P}(\mathcal{X}, \Omega|\theta)|\mathcal{X}, \theta_o\}, \tag{4}$$

where $\theta_o = [A_o, b_o]$ are the previous parameter estimates, $\mathcal{X}$ denotes the collection of observed samples x, and Ω denotes the collection of the corresponding unobserved mixture indices ω_i.

Each iteration of the EM algorithm involves an expectation (E-step) and a maximization step (M-step). In the Appendix we show that the E-step involves the computation of the sufficient statistics

$$\bar{\mu}_i = \frac{1}{n_i} \sum_x \mathrm{P}(\omega_i|A_o, b_o, x) x \tag{5}$$

$$\bar{\Sigma}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o, b_o, x)(x-\bar{\mu}_i)(x-\bar{\mu}_i)^T \quad (6)$$

$$n_i = \sum_x \mathrm{p}(\omega_i|A_o, b_o, x), \quad (7)$$

where the posterior probabilities can be computed using Bayes' rule

$$\mathrm{p}(\omega_i|A_o, b_o, x) = \frac{\mathrm{p}(\omega_i)N(x; A_o m_i + b_o, A_o S_i A_o^T)}{\sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i)N(x; A_o m_i + b_o, A_o S_i A_o^T)}. \quad (8)$$

For the one-dimensional case, and therefore for the case of diagonal covariances and a diagonal scaling matrix A, the quantities $S_i = s_i^2, A = a, \bar{\Sigma}_i = \bar{\sigma}_i^2$ and $\bar{\mu}_i, m, b$ are scalars. In this case, the M-step is equivalent to solving the following quadratic equation (see Appendix):

$$\Big(\sum_{i=1}^{N_\omega} n_i\Big)a^2 - \Big(\sum_{i=1}^{N_\omega} \frac{n_i}{s_i^2}\Big)b^2 - \Big(\sum_{i=1}^{N_\omega} \frac{n_i m_i}{s_i^2}\Big)ab + \Big(\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i m_i}{s_i^2}\Big)a + \Big(2\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i}{s_i^2}\Big)b - \Big(\sum_{i=1}^{N_\omega} n_i \frac{\bar{\mu}_i^2 + \bar{\sigma}_i^2}{s_i^2}\Big) = 0 \quad (9)$$

where the offset b is given by

$$b = \Big(\sum_{i=1}^{N_\omega} \frac{n_i \bar{\mu}_i}{s_i^2} - a\sum_{i=1}^{N_\omega} \frac{n_i m_i}{s_i^2}\Big) \Big/ \Big(\sum_{i=1}^{N_\omega} \frac{n_i}{s_i^2}\Big). \quad (10)$$

It is straightforward to verify that this equation has real roots. For the general multidimensional case—that is, when the covariances and the scaling matrix A are not diagonal—the M-step is equivalent to solving a system of second order equations. Iterative schemes may be used in the general case. In this paper, we deal only with independent constraints, that is, with diagonal covariances and scaling matrices.

2.2 Estimation of a Gaussian Mixture Density in HMMs

The constrained estimation of Gaussian mixtures can be easily extended for the dynamic case of time-varying processes with an underlying discrete Markovian state. Specifically, consider the finite-state process $[s_t, t = 1, \ldots, T]$, which can be modeled as a first-order Markov chain with transition probabilities $a_{ij} = \mathrm{p}(s_t = j|s_{t-1} = i)$. This state process

can generate an observed process $[x_t]$ through a stochastic mapping $\mathrm{p}(x_t|s_t)$, and the overall model for the process $[x_t]$ is a hidden Markov model. In the reestimation formulae for HMMs with Gaussian mixture output distributions of the form

$$\mathrm{p}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(x_t; A(g)m_i(g) + b(g), A(g)S_i(g)A^T(g)), \tag{11}$$

g is the Gaussian codebook (or genone) index. Thus, we assume that we have a collection of genones indexed by $g = 1, \ldots, N_g$, and that the mapping from HMM state s_t to genone is $g = \gamma(s_t)$. The inverse image $\gamma^{-1}(g)$ is the set of HMM states that map to the same genone (i.e., the set of HMM states that share the same mixture components). As in the static case, we assume that the parameters $m_i(g), S_i(g), i = 1, \ldots, N_\omega$ are fixed, the matrices $S_i(g)$ are positive definite, and the free parameters in the mixtures are the transformation parameters $A(g), b(g)$ which, for simplicity, are assumed to be genone-dependent.

The EM algorithm can be used to estimate the parameters of this model. The unobserved variables are the HMM state and the mixture index, and the EM algorithm in this case takes the form of the well-known Baum-Welch algorithm [3]. The formulae for the conventional reestimation of HMMs with Gaussian mixture densities can be derived by applying the Baum-Welch algorithm; see, for example, [13]. In our case, since we constrain the estimation of the Gaussians, the reestimation formulae are different, and the training procedure using the Baum-Welch algorithm is as summarized below.

1. Initialize all transformations with $A_0(g) = I, b_0(g) = 0, g = 1, \ldots, N_g$. Set $k = 0$.

2. **E-step:** Perform one iteration of the forward-backward algorithm on the speech data, using Gaussians transformed with the current value of the transformations $\theta_k(g) = [A_k(g), b_k(g)]$. For all component gaussians and all genones g collect the sufficient statistics

$$\bar{\mu}_i(g) = \frac{1}{n_i(g)} \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g)x_t \tag{12}$$

$$\bar{\Sigma}_i(g) = \frac{1}{n_i(g)} \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g)(x_t - \bar{\mu}_i(g))(x_t - \bar{\mu}_i(g))^T \tag{13}$$

$$n_i(g) = \sum_t \sum_{s_t \in \gamma^{-1}(g)} \rho(s_t)\phi_{it}(g), \tag{14}$$

where $\rho(s_t) = \mathrm{P}(s_t|\mathcal{X}, \lambda_k)$ is the probability of being at state s_t at time t given $\mathcal{X}$ and the current HMM parameters λ_k, and $\phi_{it}(g)$ is the posterior probability

$$\phi_{it}(g) = \mathrm{P}(\omega_i(g)|A_k(g), b_k(g), x_t, s_t). \tag{15}$$

3. **M-step:** Compute the new transformation parameters $[A_{k+1}(g), b_{k+1}(g)]$ using the estimation formulae (9), (10).

4. If another iteration, goto (2).

3 Application to Speaker Adaptation

3.1 Adaptation of Gaussian Codebooks

For continuous mixture-density HMMs with a large number of component mixtures it is impractical to assume that there are enough adaptation data available for independent reestimation of all the component densities. The constrained estimation that we have presented can overcome this problem, since all the components within a mixture (or a group of mixtures, if there is tying of transformations) are transformed jointly. To see how this method can be applied for adaptation, we assume that the speaker-independent (SI) HMM model for the SI vector process $[y_t]$ has observation densities of the form

$$p_{SI}(y_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(y_t; m_i(g), S_i(g)). \tag{16}$$

Adaptation of this system can be achieved by jointly transforming all the Gaussians of each genone. Specifically, we assume that, given the genone index of the HMM state s_t, the speaker-dependent vector process $[x_t]$ can be obtained by the underlying process $[y_t]$ through the transformation $x_t = A(g)y_t + b(g)$. In this case, the speaker-adapted (SA) observation densities have the form

$$p_{SA}(x_t|s_t) = \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|s_t) N(x_t; A(g)m_i(g) + b(g), A(g)S_i(g)A^T(g)), \tag{17}$$

and only the transformation parameters $A(g), b(g), g = 1, \ldots, N_g$ need to be estimated during adaptation.

The above algorithm can also be modified to monotonically approach speaker-dependent (SD) training as the amount of adaptation speech is increased. We can achieve this by setting a threshold and reestimating without constraints all individual Gaussians for which the number of samples assigned to them is larger than the threshold. Hence, all Gaussians

with a sufficiently large amount of adaptation speech are reestimated independently, whereas Gaussians with little or no adaptation data are adapted in groups. In addition, if the total amount of adaptation data for a particular genone is less than a prespecified threshold, then an identity transformation is used for all of its Gaussians.

Since our Gaussian adaptation algorithm is an instance of the Baum-Welch algorithm for HMMs with constrained mixture densities, it can be implemented efficiently. Specifically, the sufficient statistics (12) through (14) are the same as in the case of unconstrained mixture densities. Hence, the E-step at each iteration of the adaptation algorithm requires the computation and storage of these statistics and is equivalent to the E-step of the Baum-Welch algorithm for unconstrained mixture densities. The computational requirements of the M-step are very small compared to the E-step.

3.2 Adaptation of Mixture Weights

The constrained estimation algorithm that we described in the previous sections can be used to adapt the component densities of the observation distributions. Another set of parameters in a continuous-mixture HMM speech recognizer is comprised by the mixture weights $\mathrm{P}(\omega_i|s_t)$. When there is a high degree of sharing of the mixture components among different HMM states—that is, when the number of genones N_g is small—then the distributions corresponding to different HMM states are mainly distinguished by the different mixture weights. In HMMs with less sharing, as N_g increases, there is a shift in focus and the discrimination between different states is mainly achieved using the component densities. Hence, the significance of adapting the mixture weights varies, depending on the type of sharing. Since systems with a small degree of sharing usually perform better, adaptation of the Gaussians may have a greater effect on recognition performance. Nevertheless, it may

still prove beneficial to incorporate in the adaptation scheme some form of adaptation of the mixture weights.

The technique that we chose to use can be characterized as "pseudo-Bayesian". Specifically, after adapting the component Gaussians as described in Section 3.1, an additional pass through the adaptation data is performed using the forward-backward algorithm. The SD counts for the mixture weights are accumulated, and linearly combined with the SI forward-backward counts, in a fashion similar to the one reported in [10]. The weighting factor that is used determines the relative prominence given to the adaptation data. The algorithm can also be viewed as a pseudo-Bayesian adaptation scheme, where the relative contribution of the SI prior knowledge and the SD adaptation data is determined experimentally.

4 Experiments

We evaluated our adaptation algorithms on the large-vocabulary Wall Street Journal corpus [20]. Experiments were carried out using SRI's DECIPHER^{TM} speech recognition system configured with a six-feature front end that outputs 12 cepstral coefficients ($c_1 - c_{12}$), cepstral energy (c_0), and their first- and second-order differences. The cepstral features are computed from an FFT filterbank, and subsequent cepstral-mean normalization on a sentence basis is performed. We used genonic hidden Markov models with an arbitrary degree of Gaussian sharing across different HMM states as described in [8]. For fast experimentation, we used the progressive search framework [17]: an initial, speaker-independent recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using speaker-dependent or speaker-adapted models. We performed two series of experiments, on native and nonnative speakers of American English, respectively. All experiments were performed on the 5,000-word closed-vocabulary

task, and are described below.

4.1 Adaptation to Native Speakers

To compare SI, SD and SA recognition performance on native speakers, we performed an initial study of our adaptation algorithms on the phase-0 WSJ corpus. We used phonetically-tied mixture HMM systems, with all allophones of the same context-independent phone sharing the same mixture components, that is, we used systems with one genone per phone. Speaker-independent systems were trained on 3,500 sentences from 42 male speakers. The different cepstral features were modeled as independent observation streams, and each codebook used 50 Gaussians for the vector features and 15 Gaussians for the scalar (energy) features. There was a total of 6,300 phonetic models, each with three states. The number of distinct output distributions was clustered down to 6,300 (a 3-fold reduction) using state-based clustering [11], since a more compact system with fewer parameters is better suited for adaptation. The performance of the adaptation algorithm was evaluated on 100 sentences from each of six male speakers (001, 00b, 00c, 00d, 400, and 431) for varying amounts of training/adaptation sentences. The SI word error rate for these speakers was 15.51%, including deletions and insertions. We also evaluated the SD performance by separately training a speaker-dependent system for each one of the six speakers using 600 utterances, and found that the SD error rate was 11.51%. We then tested the adaptation algorithm using a small amount of adaptation data (40 utterances), and the word error rate after adaptation was 13.60%. Thus, with 40 adaptation sentences, 60% of the gap between SI and SD performance was overcome.

We then evaluated the SA system performance for varying amounts of adaptation data, using three of the speakers. The results are summarized in Figure 1. With 100 adaptation

sentences, the adaptation scheme achieves the performance of a speaker-dependent system that used 6 times as much training data. When all the SD training data are used as adaptation data, the SA system achieves a 50% reduction in error rate over the SI system and a 25% reduction over the SD system.

It is difficult to compare our work to other adaptation schemes that have appeared in the literature. The results are usually confounded by differences in:

- the task complexity. This includes vocabulary size, use of a strict language model, noise conditions, etc.
- the type of recognition system and its baseline accuracy. Systems that already exhibit a good SI performance may show small improvement due to adaptation
- the fluency of the speakers and the test-sample size. As we will see in the following section, adaptation helps nonnative speakers significantly more than native speakers.

In order to overcome some of these problems and compare our algorithm to previous work, we implemented the adaptation algorithm described in [22]. This algorithm is only suitable for tied-mixture systems: adaptation of the Gaussians is achieved using unconstrained Baum-Welch reestimation and there is no mixture-weight adaptation. We built an SI tied-mixture system and found that the SI and 40-sentence SA word error rates on the six-speaker test set were 17.0% and 16.1%, respectively. Both of these numbers are higher than the 15.5% and 13.6% word error rates that we observed using SI phonetically-tied mixtures and our adaptation algorithm.

Because of the reasons we mentioned above, we can only make qualitative comments in comparing our algorithm to previous work by others. In [16], Lee and Gauvain obtained similar SD and SA recognition performance (3.5% word error rate) with 600 sentences on the

1,000-word ARPA Resource Management (RM) task using context-independent models. Our adaptation algorithm achieved 25% lower error than SD training when 600 WSJ sentences were used. With 40 adaptation sentences, their method reduced the SI word error rate by 33% (from 6.3% to 4.2%). In our case we observed a 12% reduction. However, both of these differences may be attributed to the different domains, the amount of initial SI training data and the quality of the SI models.

Huang and Lee [10] also reported adaptation results on the RM task. They used the simple Gaussian reestimation scheme proposed by Rtischev [22] and a "pseudo-Bayesian" adaptation method for the mixture weights that is similar to the one we used in our work. On a different test set from the one used by Lee and Gauvain, they reported a 4.3% SI word error rate and a 2.6% SD word error rate using 600 SD training sentences. Their SA results were 3.6%, 2.5% and 2.4% using 40, 300 and 600 adaptation sentences, respectively. Their error rates are, in general, lower than the ones in [16]. As a consequence, Huang and Lee's error-rate reduction using 40 adaptation sentences is smaller (16%) than Lee and Gauvain's, and is comparable to ours. Also, the Huang-Lee method achieves 600-sentence SD performance after 300 adaptation sentences, and the 600-sentence SA error rate is 8% less than the corresponding SD error rate. In our case, we achieved 600-sentence SD performance after 100 adaptation sentences and our 600-sentence SA error rate is 25% lower than the corresponding SD error rate.

4.2 Adaptation to Nonnative Speakers

Speaker adaptation becomes a very important technology for outlier speakers, since the SI error rate is too high for any practical application[2]. In testing the adaptation algorithm on

[2]This was an additional motivation for all three authors of this paper, who are nonnative speakers of American English. Two of the authors are actually included in the test sets used in this section's experiments.

the "spoke 3" task of the phase-1 Wall Street Journal corpus [14], we focused on improving recognition performance for nonnative speakers of American English using adaptation. Since the phase-1 corpus was available during this series of experiments, the SI systems were built using 17,000 training utterances from 140 male speakers. To reduce computing requirements we tuned the algorithm using the five male speakers in the phase-1 WSJ development data set. The evaluation data set was run only once at the end of the development phase. The data set includes 40 test sentences and 40 phonetically balanced adaptation sentences per speaker. The speakers were selected according to their fluency in English, covering strong to light accents.

We first tested four different systems to determine the optimal degree of Gaussian sharing for this task. All of the systems used 11,932 context-dependent phonetic models, each with three states. Context dependency was modeled only within words, since we had found in preliminary experiments that modeling coarticulation across word boundaries does not improve recognition performance for nonnative speakers. The numbers of genones used in these systems were 40 (1 genone per phone), 200, 500, and 950. Each genone consisted of a mixture of 32 Gaussian distributions. The SI and SA performance is shown in Table 1. The adaptation was applied sequentially to the Gaussian distributions and the mixture weights.

In genonic HMMs, an arbitrary degree of mixture tying across different HMM states can be selected through an agglomerative clustering procedure [8]. If the degree of tying is small, and consequently the number of genones is large (as in the 500- and 950-genone systems in Table 1), then a large number of transformations may have to be estimated during adaptation. We can overcome this problem by using tying of the transformations across different genones, and the agglomerative clustering scheme used for the genone construction is very suitable for this. Each node in the tree that is generated during the clustering procedure corresponds to a set of states, with the leaves of the tree corresponding to single

HMM states. The degree of tying used in a particular system can be represented by a cut through the tree. The location of the cut is determined by the stopping criterion of the agglomerative clustering. Thus, if we want to use a smaller number of transformations than the number of genones in the system, we can somewhat relax the stopping criterion (i.e., cluster more aggressively) and determine a second cut, at a higher level through the tree. All nodes of the original cut (i.e., all genones) that fall under the same node of the new cut can share the same transformation. The third column in Table 1 indicates the number of transformations used in reestimating the Gaussian distributions. In the first two systems we used one transformation per genone. In the remaining two systems with large numbers of genones, we grouped the transformations in order to reduce the number of parameters to be estimated.

The SI word error rates for the various systems were similar, ranging from 28.7% to 30.1%. By using tying of the transformations during adaptation for the 950- and 500-genone systems and reducing the number of transformations from 950 and 500 to 200, the SA error rates were reduced from 17.7% and 16.6% to 15.8% and 15.1%, respectively. The SA error rate of 15.1% was the lowest overall for all the systems that we examined, and the average improvement due to the adaptation algorithm for the five speakers was 47%. To evaluate the relative contribution of the two stages of our adaptation scheme, we evaluated the SA error rate for our best system with the mixture-weight adaptation disabled. We found that by adapting the Gaussian codebooks only using the constrained estimation method, the SA word error rate was 15.6%. Hence, for continuous HMMs most of the performance gain during adaptation is achieved by adapting the Gaussian codebooks. Table 2 shows the results for the November 1993 ARPA evaluation set [19] on the 500-genone system. In this case the improvement is 27%.

To compare the nonnative performance before and after adaptation to that of native

speakers, we evaluated the same four systems on the same speakers that we used in Section 4.1. The results are summarized in Table 3. There we see that the SI performance of the more detailed systems (with a larger number of Gaussian distributions) is significantly better than that of the less detailed ones. This is an important difference from the nonnative results. A plausible explanation for the nonnative case is that the additional detail of the more continuous systems is not needed if the speakers are different from the training population. We also observe that for natives the SA error rate using 40 utterances is only 7% less than the SI one, as opposed to the 30% to 50% improvement that we observed for nonnatives. Moreover, the improvement is less than the 12% decrease in word error that was observed for the native speakers in the experiments with the phase-0 WSJ corpus, and is not uniform across speakers. Since the phase-1 WSJ corpus has 5 times more training data than the phase-0 corpus, we can conclude that, when a large amount of SI training data is available, adaptation is not nearly as effective for typical speakers as it is for outlier speakers.

The SI and SA word-error rates for the best systems and for both native and nonnative speakers are summarized in Table 4. The SI word error rate for nonnative speakers is 2.5 to 3 times less than that of native speakers. However, after adapting with 40 adaptation utterances, the nonnative SA error rate is approximately a factor of 1.5 higher than that of native speakers.

5 Summary

We have presented a new algorithm for the maximum-likelihood (ML) estimation of a mixture of Gaussians subject to the constraint that all means and covariances are obtained through a transformation (that needs to be estimated) from a fixed set of component densities. This constrained estimation method is well suited to the speaker adaptation

problem for continuous mixture-density HMMs with a large number of component densities that are hard to estimate in an unconstrained fashion from a small amount of adaptation data.

We tested our algorithm on the large-vocabulary WSJ corpus on both native and nonnative speakers of American English, and on a variety of recognition systems. We found that for native speakers the recognition performance after adaptation is similar to that of speaker-dependent systems that use 6 times as much training data. With small amounts of adaptation data (40 utterances with an average length of 10 seconds) the decrease in word-error rate for native speakers is approximately 7% and is much larger for nonnative speakers, ranging from 30% to 50%. This is a very important result, since the speaker-independent word-error rates for outlier speakers, like nonnative speakers, can be 2.5 to 3 times as high as those of native speakers. With speaker adaptation, outlier and nonnative speakers can use automatic speech recognition at performance levels similar to those of native speakers. Thus, the algorithm that we propose can significantly increase the usability of continuous mixture-density HMM systems. Moreover, we used the WSJ database and our results can serve as a benchmark to other researchers that want to evaluate their nonnative-speaker adaptation techniques on the same data.

We also studied the relationship between adaptation behavior and degree of mixture sharing in continuous HMM systems. We found that, with a large amount of speaker-independent training, more continuous systems with a large number of Gaussians perform better on typical native speakers in both their speaker-independent and speaker-adapted modes. However, the situation is different for atypical, nonnative speakers. For those, increasing the detail in the modeling of context dependencies is not as beneficial, since the nonnative speakers are less likely to follow the typical coarticulation patterns observed in native speakers. The result is that more compact systems actually exhibit better adaptation

performance because there are fewer parameters to adapt.

Since the results of this study are very encouraging, we are currently investigating methods to extend our adaptation algorithm to work in an unsupervised manner, that is, when the prompting text is not available for adaptation.

APPENDIX: Derivation of the Expectation and Maximization Steps

To apply the Expectation-maximization (EM) algorithm to the estimation of a Gaussian mixture, we can rewrite the auxiliary function as

$$E\{\log \mathrm{p}(\mathcal{X}, \Omega|\theta)|\mathcal{X}, \theta_o\} = \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)[\log \mathrm{p}(x|\omega_i, \theta) + \log \mathrm{p}(\omega_i|\theta)] \tag{18}$$

Since the parameters θ consist of the transformation parameters $[A, b]$, the second term in the summation does not depend on θ, and hence at each EM iteration we need to maximize the first term only.

It is well known that the joint log-likelihood of a collection of samples $\mathcal{X}$ drawn independently from a multivariate normal distribution with mean μ and covariance Σ can be expressed as [1]

$$\log \mathrm{p}(\mathcal{X}) = -\frac{n}{2}\log|\Sigma| - \frac{n}{2}(\mu - \bar{\mu})^T\Sigma^{-1}(\mu - \bar{\mu}) - \frac{n}{2}\mathrm{trace}\{\Sigma^{-1}\bar{\Sigma}\} \tag{19}$$

where $\bar{\mu}, \bar{\Sigma}$ are the sample mean and covariance, respectively, and n is the number of samples. A similar expression can be derived for the first term of the expected log-likelihood in (18). We first note that this expectation can be written

$$\begin{aligned}
\mathcal{L}(\theta; \theta_o) &= E\{\log \mathrm{p}(\mathcal{X}|\Omega, \theta)|\mathcal{X}, \theta_o\} && (20)\\
&= \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)\Big[-\frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i)\Big] && (21)\\
&= \sum_x \sum_{i=1}^{N_\omega} \mathrm{p}(\omega_i|x, \theta_o)\Big[-\frac{1}{2}\log|\Sigma_i| + \frac{1}{2}x^T\Sigma_i^{-1}\mu_i + \frac{1}{2}\mu_i^T\Sigma_i^{-1}x \\
&\quad -\frac{1}{2}x^T\Sigma_i^{-1}x - \frac{1}{2}\mu_i^T\Sigma_i^{-1}\mu_i\Big], && (22)
\end{aligned}$$

where the means and covariances are constrained $\mu_i = Am_i + b, \Sigma_i = AS_iA^T$. By expanding

the summation above, we can write

$$\begin{aligned}\mathcal{L}(\theta;\theta_o) &= \sum_{i=1}^{N_\omega}\Big[-\frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)\}\log|\Sigma_i| + \frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\}\Sigma_i^{-1}\mu_i \\ &\quad +\frac{1}{2}\mu_i^T\Sigma_i^{-1}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)x\} - \frac{1}{2}\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\Sigma_i^{-1}x \\ &\quad -\frac{1}{2}\{\sum_x \mathrm{p}(\omega_i|x,\theta_o)\}\mu_i^T\Sigma_i^{-1}\mu_i\Big]. \end{aligned} \tag{23}$$

We can define the sufficient statistics

$$n_i = \sum_x \mathrm{p}(\omega_i|A_o,b_o,x) \tag{24}$$

$$\bar{\mu}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o,b_o,x)x \tag{25}$$

$$\bar{\Sigma}_i = \frac{1}{n_i}\sum_x \mathrm{p}(\omega_i|A_o,b_o,x)(x-\bar{\mu}_i)(x-\bar{\mu}_i)^T, \tag{26}$$

and rewrite equation (23) above as

$$\begin{aligned}\mathcal{L}(\theta;\theta_o) &= \sum_{i=1}^{N_\omega}\Big[-\frac{n_i}{2}\log|\Sigma_i| - \frac{n_i}{2}(\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) + \frac{n_i}{2}\bar{\mu}_i^T\Sigma_i^{-1}\bar{\mu}_i \\ &\quad -\frac{1}{2}\sum_x \mathrm{p}(\omega_i|x,\theta_o)x^T\Sigma_i^{-1}x\Big] \qquad (27) \\ &= \sum_{i=1}^{N_\omega}\Big[-\frac{n_i}{2}\log|\Sigma_i| - \frac{n_i}{2}(\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) \\ &\quad -\frac{1}{2}\mathrm{trace}\{\Sigma_i^{-1}[\sum_x \mathrm{p}(\omega_i|x,\theta_o)xx^T - n_i\bar{\mu}_i\bar{\mu}_i^T]\}\Big] \qquad (28) \\ &= -\sum_{i=1}^{N_\omega}\frac{n_i}{2}\Big[\log|\Sigma_i| + (\mu_i-\bar{\mu}_i)^T\Sigma_i^{-1}(\mu_i-\bar{\mu}_i) + \mathrm{trace}\{\Sigma_i^{-1}\bar{\Sigma}_i\}\Big], \qquad (29)\end{aligned}$$

where in the second equation above we used the matrix identity $x^TAx = \mathrm{trace}\{Axx^T\}$ for a matrix A and a vector x, and in the third equation we used the definition of the statistic $\bar{\Sigma}_i$. The equations for the computation of the sufficient statistics comprise the E-step of the algorithm, and are summarized in (5) through (7).

To derive the M-step of the algorithm, we first rewrite Equation (29) using the transformation parameters

$$\mathcal{L}(\theta;\theta_o) = -\sum_{i=1}^{N_\omega}\frac{n_i}{2}\Big[\log|S_i| + \log|A|^2 + (A^{-1}\bar{\mu}_i - m_i - A^{-1}b)^TS_i^{-1}(A^{-1}\bar{\mu}_i - m_i - A^{-1}b)$$

$$+\text{trace}\{A^{-T}S_i^{-1}A^{-1}\bar{\Sigma}_i\}\Big], \tag{30}$$

where we have assumed that the transformation matrix A has full rank. By taking the gradient of $\mathcal{L}(\theta;\theta_o)$ with respect to the transformation parameters A, b we find the following system of equations:

$$\sum_{i=1}^{N_\omega} n_i\Big\{A - S_i^{-1}\Big[A^{-1}(\bar{\mu}_i - b) - m_i\Big](\bar{\mu}_i - b)^T - S_i^{-1}A^{-1}\bar{\Sigma}_i\Big\} = 0 \tag{31}$$

$$b = \Big[\sum_{i=1}^{N_\omega} n_i A^{-T}S_i^{-1}A^{-1}\Big]^{-1}\Big[\sum_{i=1}^{N_\omega} n_i A^{-T}S_i^{-1}A^{-1}(\bar{\mu}_i - Am_i)\Big]. \tag{32}$$

Under the assumption of diagonal covariance matrices and diagonal transformation matrices, the multidimensional case is equivalent to a set of one-dimensional problems that can be solved independently. The auxiliary function can be written in this case as

$$\mathcal{L}(\theta;\theta_o) = -\sum_{i=1}^{N_\omega}\frac{n_i}{2}[\log s_i^2 + \log a^2 + \frac{(\bar{\mu}_i - am_i - b)^2}{a^2 s_i^2} + \frac{\bar{\sigma}_i^2}{a^2 s_i^2}]. \tag{33}$$

By maximizing this quantity with respect to the transformation parameters a, b we can easily derive equations (9), (10).

Acknowledgments

We gratefully acknowledge support for this work from ARPA through Office of Naval Research Contracts N00014-93-C-0142 and N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Government funding agencies. We would also like to thank our colleagues Mike Cohen, Hy Murveit and Mitch Weintraub for their comments that improved the quality of this manuscript.

References

[1] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis,* 2nd Edition, Wiley, New York, 1984.

[2] L. R. Bahl, F. Jelinek and R. L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-5(2), pp. 179–190, March 1983.

[3] L. E. Baum, T. Petrie, G. Soules and N. Weiss, "A Maximization Technique in the Statistical Analysis of Probabilistic Functions of Finite State Markov Chains," *Ann. Math. Stat.,* Vol. 41, pp. 164–171, 1970.

[4] J. Bellegarda *et al.*, "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping," *Proceedings ICASSP,* pp. I-445–I-448, San Fransisco, CA, 1992.

[5] P. Brown, C.-H. Lee and J. Spohrer, "Bayesian Adaptation in Speech Recognition," *Proceedings ICASSP,* pp. 761–764, Boston, MA, 1983.

[6] K. Choukri, G. Chollet and Y. Grenier, "Spectral Transformations through Canonical Correlation Analysis for Speaker Adaptation in ASR," *Proceedings ICASSP,* pp. 2659–2662, Tokyo, Japan, 1986.

[7] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B)*, Vol. 39, No. 1, pp. 1–38, 1977.

[8] V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," *Proceedings ICASSP,* Adelaide, Australia, 1994.

[9] S. Furui, "Unsupervised Speaker Adaptation Method Based on Hierarchical Speaker Clustering," *Proceedings ICASSP,* pp. 286–289, Glasgow, Scotland, 1989.

[10] X. Huang and K.-F. Lee, "On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition," *IEEE Trans. on Speech and Audio Processing,* Vol. 1, No. 2, pp. 150–157, April 1993.

[11] M.-Y. Hwang and X. Huang, "Subphonetic Modeling with Markov States - Senone," *Proceedings ICASSP,* pp. I-33–36, San Fransisco, CA, 1992.

[12] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *IEEE Proceedings,* Vol. 64, No. 4, pp. 532–556, April 1976.

[13] B.-H. Juang, "Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains," *AT&T Technical Journal,* Vol.64, No.6, July-August 1985.

[14] F. Kubala *et al.*, "The Hub and Spoke Paradigm for CSR Evaluation," *Proceedings of the HLT Workshop,* Princeton, NJ, March 1994.

[15] C.-H. Lee, C.-H. Lin and B.-H. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models," *IEEE Trans. on Acoust., Speech and Signal Proc.,* Vol. ASSP-39(4), pp. 806–814, April 1991.

[16] C.-H. Lee and J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters," *Proceedings ICASSP,* pp. II-558 – II-561, Minneapolis, Minnesota, 1993.

[17] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER^{TM} Speech Recognition System: Progressive-Search Techniques," *Proceedings ICASSP,* pp. II-319 – II-322, Minneapolis, Minnesota, 1993.

[18] S. Nakamura and K. Shikano, "A Comparative Study of Spectral Mapping for Speaker Adaptation," *Proceedings ICASSP,* pp. 157–160, Albuquerque, NM, 1990.

[19] D. Pallet *et al.*, "1993 Benchmark Tests for the ARPA Spoken Language Program," *Proceedings of the HLT Workshop,* Princeton, NJ, March 1994.

[20] D. Paul and J. Baker, "The Design for the Wall Street Journal-based CSR corpus," *Proceedings of the DARPA Speech and Natural Language Workshop,* pp. 357–362, Feb. 1992.

[21] R. A. Redner and H. F. Walker, "Mixture Densities, Maximum Likelihood and the EM Algorithm," *SIAM Review,* Vol. 26, No. 2, pp. 195–239, April 1984.

[22] D. Rtischev, D. Nahamoo and M. Picheny, "Speaker Adaptation via VQ Prototype Modification," *IEEE Trans. on Speech and Audio Processing,* Vol. 2, No. 1, pp. 94–97, January 1994.

[23] R. Schwartz, Y. L. Chow and F. Kubala, "Rapid Speaker Adaptation Using a Probabilistic Spectral Mapping," *Proceedings ICASSP,* pp. 633–636, Dallas, TX, 1987.

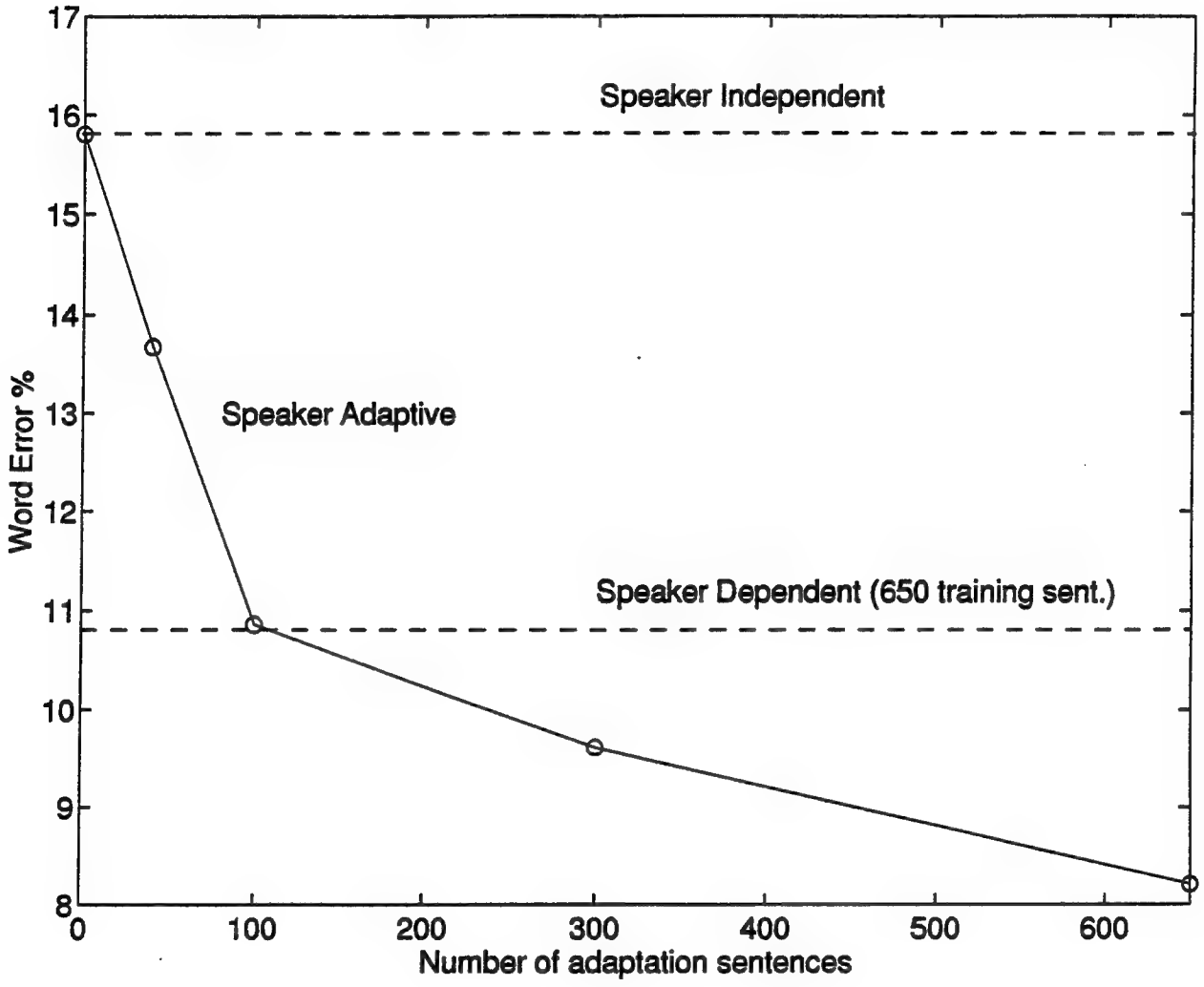

Figure 1: Speaker-independent, speaker-dependent (650 training sentences) and speaker-adaptive (varying number of sentences) word error rates for native speakers.

Speaker			4n0	4n3	4n5	4n9	4n0	AVG/SUM
Num. sentences			41	42	41	42	40	206
Num. words			719	696	664	668	678	3425
Type	Num. genones	Num. transf.						
SI	40	-	50.3	43.1	23.6	17.7	12.5	29.8
SA	40	40	24.1	18.2	17.9	12.4	9.1	16.5
SI	200	-	49.4	43.8	24.2	17.1	14.2	30.1
SA	200	200	21.4	18.7	18.4	12.0	10.5	16.2
SI	500	-	49.9	40.5	22.3	14.7	14.2	28.7
SA	500	200	20.2	15.8	16.6	12.3	10.5	15.1
SA	500	500	20.0	18.7	17.8	15.1	11.2	16.6
SI	950	-	50.5	44.7	20.5	15.3	14.4	29.5
SA	950	200	21.1	19.0	16.1	12.0	10.3	15.8
SA	950	950	24.2	21.7	18.8	13.5	9.7	17.7

Table 1: Speaker-independent (SI) and speaker-adapted (SA) word error rates for the nonnative speakers of the WSJ1 male development set for various degrees of tying and numbers of transformations.

Speaker	4nd	4ne	4nf	4ni	4nn	AVG/SUM
Num. sentences	42	42	41	41	42	208
Num. words	794	755	767	658	709	3683
SI	30.7	31.0	25.0	13.4	28.6	26.1
SA	19.0	24.5	19.7	10.2	21.0	19.1

Table 2: Word error rates for the nonnative speakers of the November 1993 WSJ1 evaluation set.

Speaker		001	00b	00c	00d	400	431	AVG/SUM
Num. sentences		50	50	50	50	50	50	300
Num. words		661	643	719	799	928	707	4457
Type	Num. genones							
SI	40	7.4	16.2	12.7	17.0	11.2	14.9	13.2
SA	40	6.7	14.8	11.3	15.3	11.0	14.1	12.3
SI	200	5.9	15.9	12.0	17.1	11.5	12.4	12.5
SA	200	6.2	16.2	13.1	13.9	10.9	12.9	12.2
SI	500	5.4	14.8	11.7	15.8	10.0	12.2	11.7
SA	500	4.8	14.8	12.0	12.8	10.0	11.3	10.9
SI	950	4.1	13.5	10.4	16.2	10.3	11.5	11.1
SA	950	3.8	13.7	11.0	12.4	9.8	10.9	10.3

Table 3: Speaker-independent (SI) and speaker-adapted (SA) word error rates for native speakers for various degrees of tying.

	SI	SA
Natives	11.1	10.3
Non natives	28.7	15.1

Table 4: Speaker-independent (SI) and speaker-adapted (SA) word error rates for native and nonnative speakers of American English.

List of Tables

List of Figures

Submitted 12/94 to the IEEE Transactions on Speech & Audio Processing

Speaker Adaptation Using Combined Transformation and Bayesian Methods

V. Digalakis 415-859-5540 vas@speech.sri.com

L. Neumeyer 415-859-4522 leo@speech.sri.com

SRI International
333 Ravenswood Ave., Menlo Park, CA 94025
Fax: 415-859-5984
November 23, 1994

EDICS SA 1.6.7

ABSTRACT

Adapting the parameters of a statistical speaker-independent continuous-speech recognizer to the speaker and the channel can significantly improve the recognition performance and robustness of the system. In continuous mixture-density hidden Markov models the number of component densities is typically very large, and it may not be feasible to acquire a sufficient amount of adaptation data for robust maximum-likelihood estimates. To solve this problem, we have recently proposed a constrained estimation technique for Gaussian mixture densities. To improve the behavior of our adaptation scheme for large amounts of adaptation data, we combine it here with Bayesian techniques. We evaluate our algorithms on the large-vocabulary Wall Street Journal corpus for nonnative speakers of American English. The recognition error rate is approximately halved with only a small amount of adaptation data, and it approaches the speaker-independent accuracy achieved for native speakers.

1 INTRODUCTION

Automatic speech recognition performance degrades rapidly when there is a mismatch between the testing and the training conditions, under which the recognizer parameters were estimated. It may not always be feasible to have consistent conditions in the testing and training phases. For example, in large-vocabulary dictation applications the speaker-independent performance degrades dramatically for outlier speakers, such as nonnative speakers of the recognizer language. Since modern large-vocabulary speech recognizers have millions of free parameters, it is not practical to collect large amounts of speaker-dependent data and retrain the recognizer models. Similarly, it is desirable to avoid the expense of collecting additional data when the recognizer is going to be used on speech transmitted through a different channel than the one used in training. Such problems may be solved by adapting the recognizer models, using much smaller amounts of adaptation data than those used in conventional training techniques. In this paper we focus on adapting the models to the speaker, although the techniques we describe can also be used at other levels [1].

One family of adaptation approaches attempts to match the new speaker's observations to the speaker-independent training data by transforming the new speaker's feature space [2][3][4]. The transformation approach has the advantage of simplicity. In addition, if the number of free parameters is small, then transformation techniques adapt to the user with only a small amount of adaptation speech (quick adaptation). A disadvantage of transformation methods is that they are usually text-dependent, that is, the new speaker must record sentences with the same text recorded previously by some reference speakers. Moreover, transformation methods may not take full advantage of large amounts of adaptation data.

A second family of adaptation algorithms follows a Bayesian approach, where the speaker-independent information is encapsulated in the prior distributions [5][6]. The Bayesian approach is text-independent, and has the nice property that speaker-adaptive performance will converge to speaker-dependent performance as the amount of adaptation speech increases. However, the adaptation rate is usually slow.

In this paper we present adaptation schemes that combine the quick adaptation characteristics of transformation-based methods with the nice asymptotic properties of Bayesian methods. We first present a transformation-based method for continuous mixture-density hidden Markov models (HMMs) that was introduced in [7]. Adaptation is achieved via a transformation of the speaker-independent observation densities, and the transformation parameters are obtained using the maximum-likelihood (ML) criterion. The number of transformation parameters can be adjusted based on the available amount of adaptation data for quick adaptation. We then show how this algorithm can be combined with Bayesian techniques. The combined method adapts to a new speaker with small amounts of adaptation data and takes better advantage of large amounts of adaptation data than the transformation method.

2 TRANSFORMATION-BASED ADAPTATION

Transformation-based approaches to speaker adaptation are typically text-dependent, that is they require the new speaker to record some utterances with predetermined text. These utterances are aligned to ones recorded by reference speakers, and mappings between the new-speaker and the reference-speaker acoustic spaces are obtained using regression techniques [3][4][8].

In [7] we presented a novel transformation-based approach to speaker adaptation for continuous mixture-density HMMs. To eliminate mismatched training and testing conditions, transformations can be applied either directly to the features, or to the speech models [9]. We chose to apply the transformation at the distribution level, rather than transforming the feature vectors directly, since we can then use the Expectation-Maximization (EM) algorithm [10] to estimate the transformation parameters by maximizing the likelihood of the adaptation data (see Figure 1a). One advantage of this approach is that the need for time alignment between new and reference speaker data is eliminated, and the transformation parameters can be estimated using new-speaker data alone. The estimation of the transformation can also be viewed as a constrained estimation of Gaussian mixtures.

For continuous mixture-density HMMs with a large number of component mixtures, it is impractical to assume that enough adaptation data are available for independent reestimation of all the component densities. The constrained estimation we presented in [7] overcomes this problem by applying the same transformation to all components of a particular mixture (or a group of mixtures, if there is tying of transformations). Gaussians for which there were no observations in the training data are adapted based on data that were most likely generated by other Gaussians of the same or other neighboring mixtures.

To see how this method can be applied for adaptation, we assume that the speaker-independent (SI) HMM model for the SI vector process $[y_t]$ has observation densities of the form

$$p_{SI}(y_t \mid s_t) = \sum_i p(\omega_i \mid s_t) N(y_t; \mu_{ig}, \Sigma_{ig}) \quad , \tag{1}$$

where g is the index of the Gaussian codebook used by state s_t.

Adaptation of this system can be achieved by jointly transforming all the Gaussians of each mixture. Specifically, we assume that, given the HMM state s_t, the speaker-dependent vector process $[x_t]$ can be obtained by an underlying process $[y_t]$ through the transformation

$$x_t = A_g y_t + b_g \quad . \tag{2}$$

Under this assumption, the speaker-adapted (SA) observation densities will have the form

$$p_{SA}(x_t \mid s_t) = \sum_i p(\omega_i \mid s) N\left(x_t; A_g \mu_{ig} + b_g, A_g \Sigma_{ig} A_g^T\right) \tag{3}$$

and only the parameters A_g, b_g, $g = 1, \ldots, N$ need to be estimated during adaptation, where N is the number of distinct transformations. The same transformations can be applied to different HMM states, and this tying of transformations can be used to optimize performance based on the amount of available adaptation data. The transformation parameters can be estimated using the EM algorithm. The reestimation formulae for the transformation parameters are derived in [7] and are summarized below.

1. Initialize all transformations with $A_g(0) = I, b_g(0) = 0, g = 1, \ldots, N$. Set $k=0$.

2. **E-step:** Perform one iteration of the forward-backward algorithm on the speech data, using Gaussians transformed with the current value of the transformations $A_g(k), b_g(k)$. For all component Gaussians and all mixtures g, collect the sufficient statistics

$$\Pi_{ig} = \frac{1}{n_{ig}} \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)\, x_t$$

$$\Sigma_{ig} = \frac{1}{n_{ig}} \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)\, (x_t - \Pi_{ig})\, (x_t - \Pi_{ig})^T \qquad (4)$$

$$n_{ig} = \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)$$

where $\gamma_t(s_t)$ is the probability of being at state s_t at time t given the current HMM parameters, the summation is over all times and HMM states that share the same mixture components, and $\phi_{it}(s_t)$ is the posterior probability

$$\phi_{it}(s_t) = p(\omega_{ig} \mid A_g(k), b_g(k), x_t, s_t) \quad . \qquad (5)$$

3. **M-step:** Compute the new transformation parameters. Under the assumption of diagonal covariance and transformation matrices, the elements a and b of $A_g(k+1), b_g(k+1)$ can be obtained by solving the following equations for each g

$$\left(\sum_i n_i\right) a^2 - \left(\sum_i \frac{n_i}{\sigma_i^2}\right) b^2 - \left(\sum_i \frac{n_i \mu_i}{\sigma_i^2}\right) ab + \left(\sum_i \frac{n_i \Pi_i \mu_i}{\sigma_i^2}\right) a + \left(2 \sum_i \frac{n_i \Pi_i}{\sigma_i^2}\right) b - \left(\sum_i n_i \frac{\Pi_i^2 + \sigma_i^2}{\sigma_i^2}\right) = 0 \qquad (6)$$

$$b = \left(\sum_i \frac{n_i \Pi_i}{\sigma_i^2} - a \sum_i \frac{n_i \mu_i}{\sigma_i^2}\right) \Big/ \left(\sum_i \frac{n_i}{\sigma_i^2}\right)$$

where for simplicity we have dropped the dependence on g. The variables μ_i, σ_i^2, Π_i, $\bar{\sigma}_i^2$ are elements of the vectors and diagonal matrices μ_{ig}, Σ_{ig}, Π_{ig}, Σ_{ig}, respectively.

4. If the convergence criterion is not met, go to step 2.

Once the transformation parameters are determined, the constrained ML estimates for the means and covariances can be obtained using

$$\mu_{ig}^{CML} = A_g \mu_{ig} + b_g$$
$$\Sigma_{ig}^{CML} = A_g \Sigma_{ig} A_g^T \qquad (7)$$

3 COMBINING TRANSFORMATION AND BAYESIAN-BASED ADAPTATION

In Bayesian adaptation techniques the limited amount of adaptation data is optimally combined with the prior knowledge. With the appropriate choice of the prior distributions, the maximum *a posteriori* (MAP) estimates for the means and covariances of HMMs with single-Gaussian observation densities can be obtained using linear combinations of the speaker-dependent counts and some quantities that depend on the parameters of the prior distributions [5]. We use the term *counts* above to denote the sufficient statistics collected by performing one iteration of the forward-backward algorithm on the adaptation data. MAP estimates for the parameters of continuous mixture-density HMMs can be obtained in the same way, as shown in [6]. For example, the MAP estimate for the mean of the *i*th Gaussian in the HMM mixture density of the *g*th Gaussian codebook can be obtained using [6]

$$\mu_{ig}^{MAP} = \frac{\tau_{ig} m_{ig} + \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)\, x_t}{\tau_{ig} + \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)} \qquad (8)$$

where $\gamma_t(s_t)$ is the probability of being at state s_t at time t given the current HMM parameters, and $\phi_{it}(s_t)$ is the posterior probability of the ith mixture component

$$\phi_{it}(s) = p(\omega_{ig} \mid x_t, s_t) = \frac{p(\omega_{ig} \mid s_t)\, N(x_t; \mu_{ig}, \Sigma_{ig})}{\sum_j p(\omega_{jg} \mid s_t)\, N(x_t; \mu_{jg}, \Sigma_{jg})} \qquad (9)$$

The quantities τ_{ig}, m_{ig} are parameters of the joint prior density of the mixture parameters, which was chosen in [6] as a product of the Dirichlet and normal-Wishart densities. The parameter τ_{ig} is usually estimated empirically and can be used to control the adaptation

rate. Similar estimation formulae can be used for the covariances of the Gaussians. Based on (8) and the similar formulae for the second-order statistics, an approximate MAP (AMAP) estimation scheme can be implemented by linearly combining the speaker-independent and the speaker-dependent counts (see Figure 1b) for each component density

$$\begin{aligned} \langle x \rangle_{ig}^{AMAP} &= \lambda \langle x \rangle_{ig}^{SI} + (1-\lambda) \langle x \rangle_{ig}^{SD} \\ \langle xx^T \rangle_{ig}^{AMAP} &= \lambda \langle xx^T \rangle_{ig}^{SI} + (1-\lambda) \langle xx^T \rangle_{ig}^{SD}, \\ n_{ig}^{AMAP} &= \lambda n_{ig}^{SI} + (1-\lambda) n_{ig}^{SD} \end{aligned} \tag{10}$$

where the superscripts on the right-hand side denote the data over which the following statistics (counts) are collected during one iteration of the forward-backward algorithm

$$\begin{aligned} \langle x \rangle_{ig} &= \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)\, x_t \\ \langle xx^T \rangle_{ig} &= \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t)\, x_t x_t^T. \\ n_{ig} &= \sum_{t, s_t} \gamma_t(s_t)\, \phi_{it}(s_t) \end{aligned} \tag{11}$$

The weight λ controls the adaptation rate. Using the combined counts, we can compute the AMAP estimates of the means and covariances of each Gaussian component density from

$$\begin{aligned} \mu_{ig}^{AMAP} &= \frac{\langle x \rangle_{ig}^{AMAP}}{n_{ig}^{AMAP}} \\ \Sigma_{ig}^{AMAP} &= \frac{\langle xx^T \rangle_{ig}^{AMAP}}{n_{ig}^{AMAP}} - \mu_{ig}^{AMAP} \left(\mu_{ig}^{AMAP} \right)^T \end{aligned} \tag{12}$$

Similar adaptation schemes have also appeared for discrete HMMs [11], and can be used to adapt the mixture weights in the approximate Bayesian scheme described here.

In Bayesian adaptation schemes, only the Gaussians of the speaker-independent models that are most likely to have generated some of the adaptation data will be adapted to the speaker. These Gaussians may represent only a small fraction of the total number in con-

tinuous HMMs with a large number of Gaussians. On the other hand, as the amount of adaptation data increases, the speaker-dependent statistics will dominate the speaker-independent priors and Bayesian techniques will approach speaker-dependent performance. We should, therefore, aim for an adaptation scheme that retains the nice properties of Bayesian schemes for large amounts of adaptation data, and has improved performance for small amounts of adaptation data. We can achieve this by using our transformation-based adaptation as a preprocessing step to transform the speaker-independent models so that they better match the new speaker characteristics and improve the prior information in MAP estimation schemes. To combine the transformation and the approximate Bayesian methods, we can first transform the speaker-independent counts using the transformation parameters estimated with the constrained ML method described in Section 2,

$$\begin{aligned} \langle x \rangle_{ig}^{CML} &= A_g \langle x \rangle_{ig}^{SI} + b_g \\ \langle xx^T \rangle_{ig}^{CML} &= A_g \langle xx^T \rangle_{ig}^{SI} A_g^T + A_g \langle x \rangle_{ig}^{SI} b_g^T + b_g \langle x^T \rangle_{ig}^{SI} A_g^T + n_{ig}^{SI} b_g b_g^T \end{aligned} \tag{13}$$

The transformed counts can then be combined with the speaker-dependent counts collected using the adaptation data

$$\begin{aligned} \langle x \rangle_{ig}^{COM} &= \lambda \langle x \rangle_{ig}^{CML} + (1-\lambda) \langle x \rangle_{ig}^{SD} \\ \langle xx^T \rangle_{ig}^{COM} &= \lambda \langle xx^T \rangle_{ig}^{CML} + (1-\lambda) \langle xx^T \rangle_{ig}^{SD} \\ n_{ig}^{COM} &= \lambda n_{ig}^{CML} + (1-\lambda) n_{ig}^{SD} \end{aligned} \tag{14}$$

and the combined-method models can be estimated from these counts using

$$\begin{aligned} \mu_{ig}^{COM} &= \frac{\langle x \rangle_{ig}^{COM}}{n_{ig}^{COM}} \\ \Sigma_{ig}^{COM} &= \frac{\langle xx^T \rangle_{ig}^{COM}}{n_{ig}^{COM}} - \mu_{ig}^{COM} \left(\mu_{ig}^{COM} \right)^T \end{aligned} \tag{15}$$

This procedure is shown schematically in Figure 1c.

4 EXPERIMENTAL RESULTS

We evaluated our adaptation algorithms on the Spoke 3 task of the phase-1, large-vocabulary Wall Street Journal (WSJ) corpus [12][13], trying to improve recognition performance for nonnative speakers of American English. Each test set used in this section consists of ten nonnative speakers of American English whose first languages are broadly distributed across the major languages. Experiments were carried out using SRI's DECIPHER^{TM} speech recognition system configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from a fast Fourier transform (FFT) filterbank, and subsequent cepstral-mean normalization on a sentence basis is performed. We used genonic hidden Markov models with an arbitrary degree of Gaussian sharing across different HMM states as described in [11]. The speaker-independent continuous HMM systems that we used as seed models for adaptation were gender-dependent, trained on 140 speakers and 17,000 sentences for each gender. Each of the two systems had 12,000 context-dependent phonetic models that shared 500 Gaussian codebooks with 32 Gaussian components per codebook. For fast experimentation, we used the progressive search framework [15]: an initial, speaker-independent recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using speaker-adapted models. We used the baseline 5,000-word, closed-vocabulary[1] bigram and trigram language models provided by the MIT Lincoln Laboratory. The trigram language model was implemented using the N-best rescoring paradigm [16], by rescoring the list of the N-best sentence hypotheses generated using the bigram language model.

In the first series of experiments we used the bigram language model. We first evaluated the performance of the transformation-based adaptation for various numbers of transformations and amounts of adaptation data. As we can see in Figure 2, where we have plotted the word error rate as a function of the number of adaptation sentences, multiple transformations outperform very constrained schemes that use 1 or 2 transformations. The perfor-

1. A closed-vocabulary language model is intended for recognizing speech that does not include words outside of the vocabulary.

mance with 20 and 40 transformations is similar, and is better than the less constrained case of 160 transformations. However, as the amount of adaptation data increases, the 160 transformations take advantage of the additional data and outperform the more constrained schemes. A significant decrease in error rate is obtained with as little as 5 adaptation sentences. When adapting using a single sentence, the performance is similar for different numbers of transformations, except for the case of two transformations. The reason is that in our implementation a transformation is reestimated only if the number of observations is larger than a threshold; otherwise, we use a global transformation estimated from all data. Since most of the transformations are backed off to the global transformation for the case of a single adaptation sentence, the cases with different numbers of transformations exhibit similar performance.

In Figure 3 we have plotted the word error rates of the combined scheme for the same numbers of transformations and adaptation sentences as in Figure 2. The systems used to obtain the results of Figure 2 are used as priors for the subsequent Bayesian estimation step, as explained in Section 3. We can see that the performance of the combined scheme becomes less sensitive to the number of transformations used, especially with larger numbers of adaptation sentences. This behavior should be expected, since Bayesian schemes will asymptotically converge to speaker-dependent performance as the amount of adaptation data increases. However, when the number of adaptation sentences is small, it is important to select the appropriate number of transformations and provide the Bayesian step with good prior information.

In Figure 4 we compare the word error rates of the transformation-only method with 20 and 160 transformations, the approximate Bayesian method with conventional priors, and the combined method for various amounts of adaptation data. In the latter, the number of transformations was optimized on an independent test set according to the available amount of adaptation data. The transformation-only method with 20 transformations outperforms the Bayesian scheme with conventional priors when fewer than 10 sentences are used for adaptation, whereas the situation reverses as more adaptation sentences are used. This is consistent with our claim that transformation-based methods adapt faster, whereas

Bayesian schemes have better asymptotic properties. The performance of the transformation approach for large amounts of adaptation data can be improved by increasing the number of transformations. In the same figure, we can also see the success of the combined method, which outperforms significantly the first two methods over the whole range of adaptation sentences that we examined. The transformation step provides quick adaptation when few adaptation sentences are used, and the Bayesian reestimation step improves the asymptotic performance.

Finally, we evaluated the word error rate of our best-performing configuration on the 1993 Spoke-3 development and evaluation sets, and the 1994 evaluation set of the WSJ corpus using a trigram language model. Our results for the 1993 test sets, presented in Table 1, represent the best reported results to date on this task [17][2]. The speaker-independent word error rate for nonnative speakers is reduced by a factor of 2 using only 40 adaptation sentences. Using 200 adaptation sentences, the speaker-adapted error rate of nonnative speakers for the November 1994 test set is 8.2%. This number is comparable to the speaker-independent word error rate of the same recognition system on the 1993 development and 1994 evaluation sets of native speakers, which is 7.2% and 8.1%, respectively.

The improvement in performance is bigger for certain outlier speakers. The first author of this paper is a nonnative speaker of American English with a particularly heavy accent. His adaptation results for as many as 285 adaptation sentences (approximately 40 minutes of speech) are summarized in Table 2, where we see that his speaker-independent error rate decreases by a factor of 4 and 6 using 40 and 285 adaptation sentences, respectively. His speaker-adapted error rate of 7.1% is comparable to the state-of-the-art performance for native speakers on this task.

5 SUMMARY

We combined the transformation-based adaptation algorithm that we presented in [7] with Bayesian methods. We presented experiments that compare the performance of transfor-

2. The 1994 official ARPA benchmark results were not available when this paper was written.

mation and Bayesian adaptation for various amounts of adaptation data. Transformation-based adaptation performs well when only a limited amount of adaptation data is available, but Bayesian methods are better as the amount of adaptation data increases. The combined method retains the quick adaptation characteristics of transformation methods, and takes advantage of the nice asymptotic properties of Bayesian schemes as the amount of adaptation data increases. The combined scheme clearly outperforms both Bayesian and transformation methods over the whole range of various amounts of adaptation speech that we examined. Our baseline results are the best reported to date on the nonnative-speaker task of the Wall Street Journal corpus, and our nonnative speaker-adapted performance is comparable to the native speaker-independent performance with sufficient amounts of adaptation speech.

Acknowledgments

This research was supported by the Advanced Research Projects Agency through Office of Naval Research Contracts ONR N00014-92-C-0154 and ONR N00014-93-C-0142. The Government has certain rights in this material. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Project Agency.

REFERENCES

[1] L. Neumeyer and M. Weintraub, "Robust Speech Recognition in Noise Using Mapping and Adaptation Techniques," to appear in *Proceedings ICASSP,* May 1995.

[2] J. Bellegarda, P. V. de Souza, A. J. Nadas, D. Nahamoo, M. A. Picheny, and L. R. Bahl, "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping," *Proceedings ICASSP,* pp. I-445—I-448, San Fransisco, CA, 1992.

[3] K. Choukri, G. Chollet and Y. Grenier, "Spectral Transformations through Canonical Correlation Analysis for Speaker Adaptation in ASR," *Proceedings ICASSP,* pp. 2659—2662, Tokyo, Japan, 1986.

[4] S. Nakamura and K. Shikano, "A Comparative Study of Spectral Mapping for Speaker Adaptation," *Proceedings ICASSP,* pp. 157—160, Albuquerque, NM, 1990.

[5] C.-H. Lee, C.-H. Lin and B.-H. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models," *IEEE Trans. on Acoust., Speech and Signal Proc.,* Vol. ASSP-39(4), pp. 806—814, April 1991.

[6] C.-H. Lee and J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters," *Proceedings ICASSP,* pp. II-558—II-561, Minneapolis, Minnesota, 1993.

[7] V. Digalakis, D. Rtischev and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," submitted to *IEEE Trans. on Speech and Audio Processing,* April 1994.

[8] R. Schwartz, Y. L. Chow and F. Kubala, "Rapid Speaker Adaptation Using a Probabilistic Spectral Mapping," *Proceedings ICASSP,* pp. 633—636, Dallas, TX, 1987.

[9] A. Sankar and C.-H. Lee, "Stochastic Matching for Robust Speech Recognition," *IEEE Signal Processing Letters,* Vol.1, No.8, August 1994.

[10] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B),* Vol. 39, No. 1, pp. 1—38, 1977.

[11] X. Huang and K.-F. Lee, "On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 1, No. 2, pp. 150—157, April 1993.

[12] F. Kubala *et al.*, "The Hub and Spoke Paradigm for CSR Evaluation," *Proceedings of the HLT workshop, Princeton, NJ, March 1994.*

[13] D. Paul and J. Baker, "The Design for the Wall Street Journal-based CSR corpus," *Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 357—362, Feb. 1992.

[14] V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," *Proceedings ICASSP*, Adelaide, Australia, 1994.

[15] H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques," *Proceedings ICASSP*, pp. II-319—II-322, Minneapolis, Minnesota, 1993.

[16] R. Schwartz and Y.-L. Chow, "A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses," *Proc. ICASSP*, pp. 701-704, May 1991.

[17] D. Pallet, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, and M. A. Przybocki, "1993 Benchmark Tests for the ARPA Spoken Language Program," *Proceedings of the HLT Workshop*, Princeton, NJ, March 1994.

TABLES

Test Set	# of Adaptation Sentences	Speaker-independent rate (%)	Speaker-adapted rate (%)
Development 93	40	23.5	10.3
Evaluation 93	40	16.5	10.0
Evaluation 94	40	23.2	11.3
	100		9.4
	200		8.2

TABLE 1. Speaker-independent and speaker-adapted word error rates on various test sets of nonnative speakers using different amounts of adaptation data.

System	# of Adaptation Sentences	Speaker-adapted rate (%)
Speaker Independent	0	42.7
Speaker Adapted	40	10.6
	285	7.1

TABLE 2. Word error rates for development speaker 4n0 and various amounts of adaptation data

FIGURES

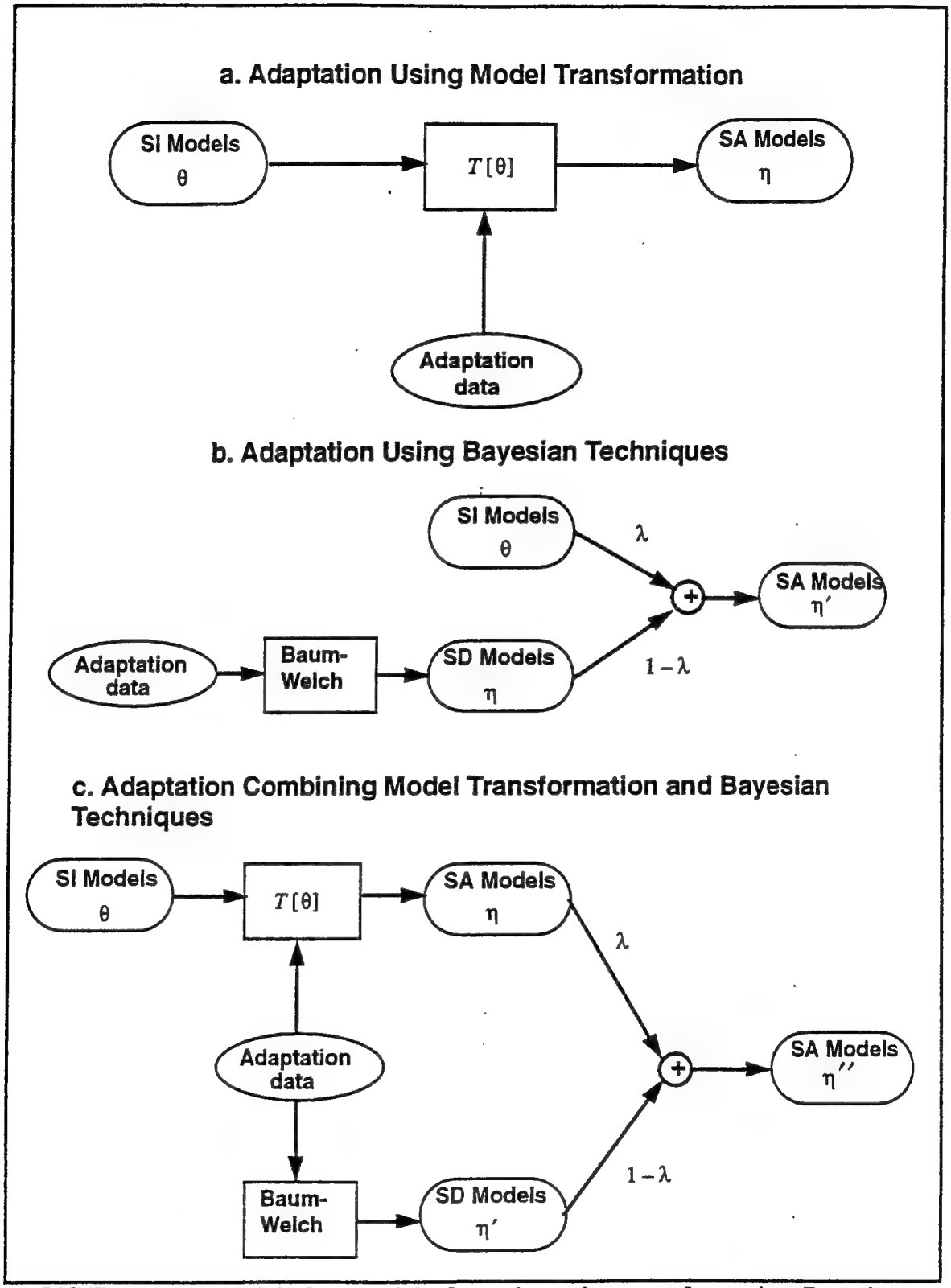

FIGURE 1. Hidden Markov model adaptation using transformation, Bayesian and combined techniques

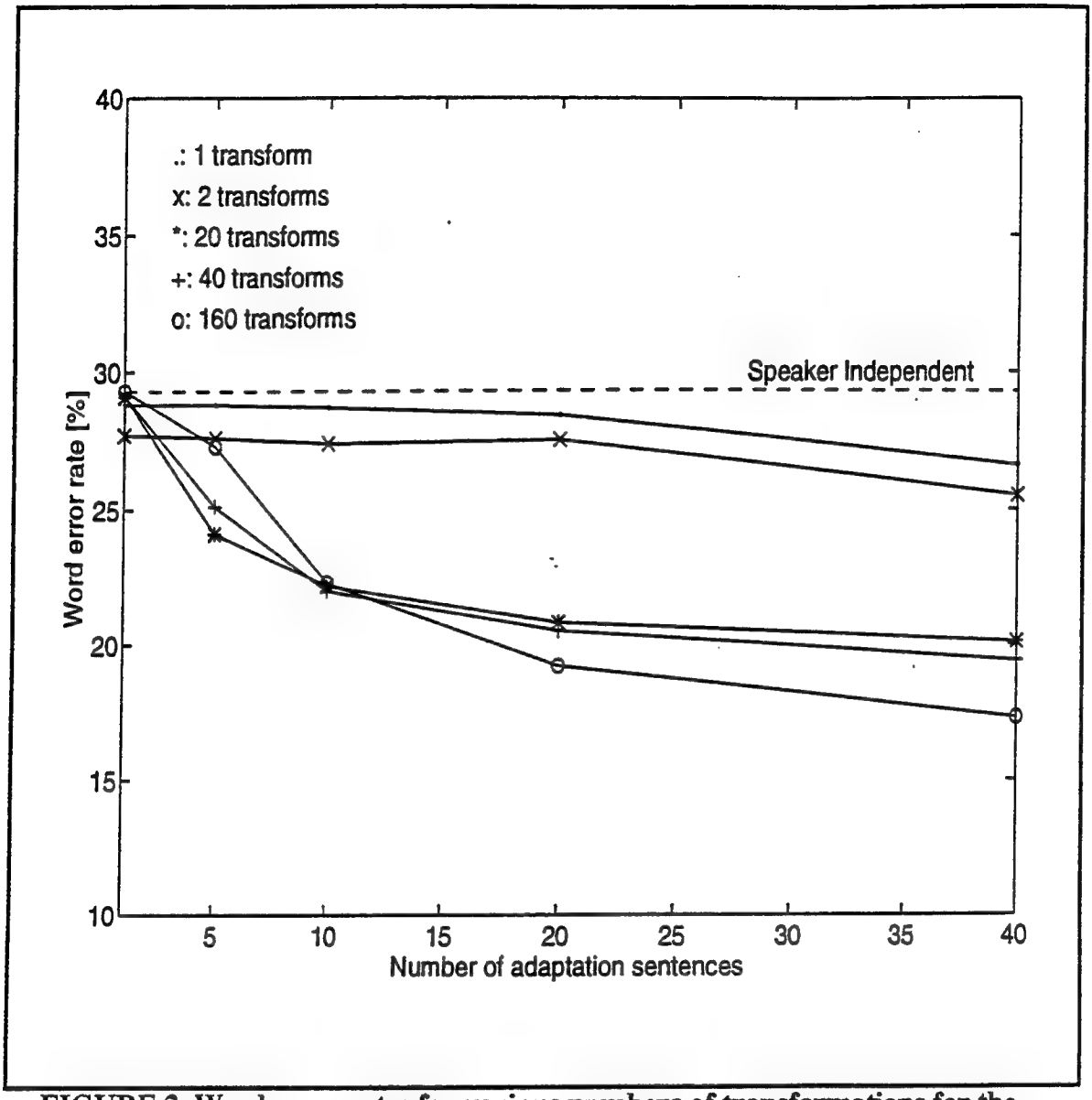

FIGURE 2. Word error rates for various numbers of transformations for the transformation-based adaptation

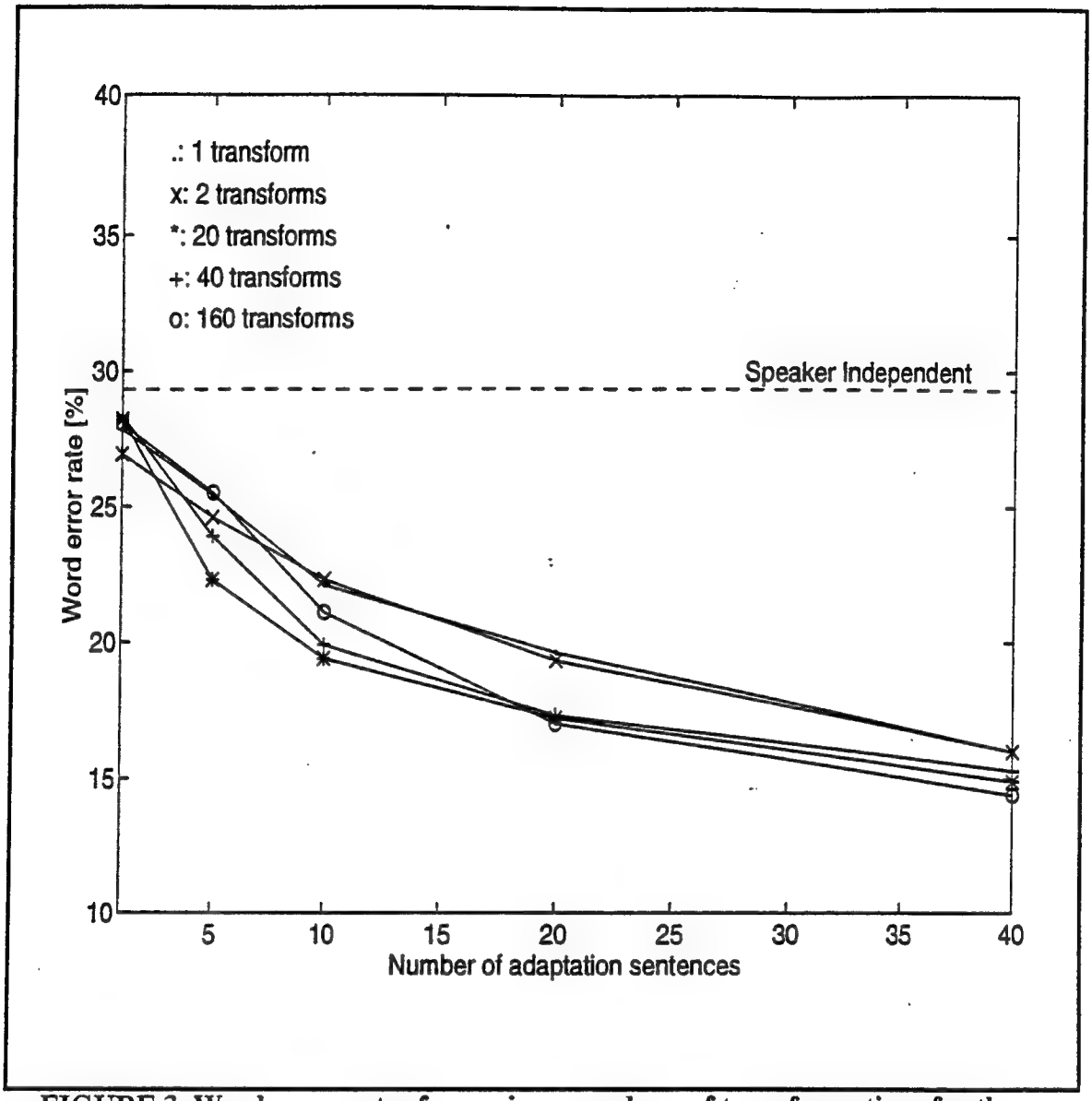

FIGURE 3. Word error rates for various numbers of transformations for the combined method

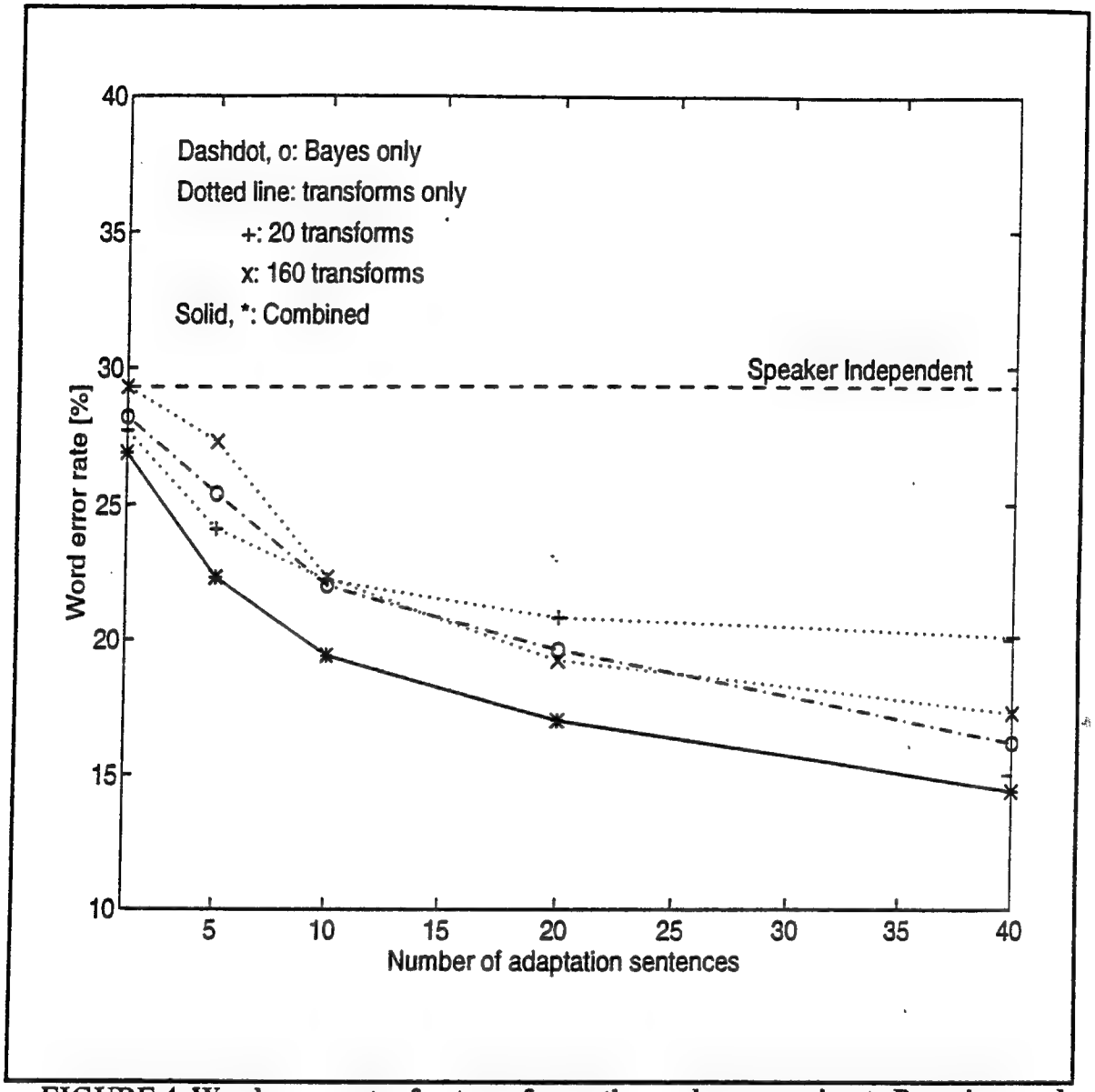

FIGURE 4. Word error rates for transformation-only, approximate Bayesian, and combined schemes

Training Issues and Channel Equalization Techniques for the Construction of Telephone Acoustic Models Using a High-Quality Speech Corpus

Leonardo G. Neumeyer, *Member, IEEE*, Vassilios V. Digalakis, and Mitchel Weintraub

***Abstract*— We describe an approach for the estimation of acoustic phonetic models that will be used in a hidden Markov model (HMM) recognizer operating over the telephone. We explore two complementary techniques to developing telephone acoustic models. The first technique presents two new channel compensation algorithms. Experimental results on the *Wall Street Journal* corpus show no significant improvement over sentence-based cepstral-mean removal. The second technique uses an existing "high-quality" speech corpus to train acoustic models that are appropriate for the *Switchboard Credit Card* task over long-distance telephone lines. Experimental results show that cross-database acoustic training yields performance similar to that of conventional task-dependent acoustic training.**

I. Introduction

In many practical situations, an automatic speech recognizer has to operate in various but well-defined acoustic environments. The training corpus, however, is usually recorded with acoustic conditions that may not exactly match those encountered in the field. This mismatch between the acoustics of the training and testing data will degrade the accuracy of the recognizer. To overcome the data mismatch problem without collecting a new training corpus for each acoustic environment, we need a representation of the speech signal that is invariant across the acoustic spaces. Our purpose is to evaluate different techniques that facilitate the construction of acoustic models for speech recognition applications over a telephone channel.

The traditional approach to building speech recognizers is to collect training data under conditions that match as closely as possible the environment in which the recognizer will be used. To attain the best possible recognition performance, researchers typically try to match the language characteristics and acoustic environment in the training and testing phases. However, if there is no mismatch between the language characteristics of the training and testing data, then one can alternatively use algorithms to correct the *acoustic* mismatch between the training and testing corpora. This approach eliminates the need to collect speech data for each new acoustic environment. We will follow a twofold algorithmic approach to the acoustic mismatch problem. We first use a channel equalization algorithm that minimizes the channel mismatch between training and testing. We will compare a number of different equalization algorithms that remove some of the simplifying assumptions in the widely used sentence-based cepstral-mean removal, and show that the simple cepstral-mean removal algorithm is highly effective in correcting channel distortions. Once the channel distortion is reduced, our second main goal is to select a front end that is suitable for the testing conditions. In telephone applications, for example, the spectral bandwidth of the channel is limited to 3 kHz. Most of the spectral energy and the relevant information required for speech recognition are contained in this range. Hence, limiting the bandwidth can only increase the robustness of the recognizer to channel distortions and background noise. We show that by designing an appropriate front end for telephone-bandwidth speech, we are able to achieve with cross-database training similar performance to task-dependent training.

The remainder of this paper is organized as follows. In Section II, we explore two different channel normalization algorithms. The first algorithm performs cepstral normalization in the log-DFT domain rather than in the log-filterbank domain. The second algorithm jointly estimates the channel and the HMM parameters during training, and the channel and most likely HMM state sequence during recognition. The performance of these two equalization algorithms is similar to the cepstral-mean removal algorithm on the alternate-microphone task of the *Wall Street Journal (WSJ)* corpus [1]. In Section III, we discuss techniques to train acoustic models with data recorded with a high-quality Sennheiser microphone for use over the telephone.

II. Channel Equalization

Although cepstral-mean normalization (CMN) is a simple technique that has been effectively used for convolutional noise removal [2], it still entails a few simplifying assumptions. In this section we present two novel algorithms that remove these assumptions.

A. Spectral Equalization in the Log DFT Domain

We first compare CMN to a different approach for the removal of stationary convolutional noise, "log-DFT mean normalization" (LDMN), and show that CMN is suboptimal when the cepstrum is computed as a linear transformation of the filterbank log energies. Specifically, we show that CMN can remove stationary convolutional noise only when

Manuscript received November 18, 1993; revised April 8, 1994. This work was supported by the Advanced Research Projects Agency under Contract ONR N00014-93-C-0142 and ONR N00014-92-C-1054, by Grant NSF IRI-9014829 from the National Science Foundation, and by SRI International internal research and development funds.

The authors are with SRI International, Menlo Park, CA 94025 USA.

IEEE Log Number 9403967.

the magnitude of the DFT of the channel's impulse response is constant in each spectral band of the filterbank. We also show that we can overcome this assumption by equalizing the spectrum in the log-DFT domain.

In a filterbank-based front end, the DFT energies are integrated to compute the mel-filterbank energies. The log filterbank energies are used to compute the mel-cepstrum, which is normalized by removing its mean in each sentence.

Consider the following speech signal corrupted with stationary convolutional noise

$$y[t] = x[t]^* h[t] \tag{1}$$

where $x[t]$ is the clean speech sequence, $h[t]$ is the impulse response of the channel, and $y[t]$ is the distorted speech. After applying the Discrete Fourier Transform to a frame[1] of speech, we get the spectral energy equation,

$$Y_{k,n} = X_{k,n} H_k \tag{2}$$

where k is the DFT index and n is the frame index. The log filterbank energy is given by

$$\log F_{l,n} = \log \sum_k w_{k,l} X_{k,n} H_k \tag{3}$$

where $F_{l,n}$ is the filterbank energy for band l in frame n and $w_{k,l}$ is a filter weight coefficient (this coefficient is zero outside the spectral band of the filter). If we assume that H_k is constant within the frequency band l

$$H_k = \tilde{H}_l \quad \forall k: w_{k,l} \neq 0 \tag{4}$$

we can express the log filterbank energy as follows:

$$\begin{aligned} \log F_{l,n} &\cong \log \left(\tilde{H}_l \sum_k w_{k,l} X_{k,n} \right) \\ &= \log \tilde{H}_l + \log \sum_k w_{k,l} X_{k,n} \end{aligned} \tag{5}$$

and the constant term $\log \tilde{H}_l$ is eliminated with cepstral mean subtraction.

To avoid the approximation in (4), we can simply normalize the spectrum in the log-DFT domain before the filterbank integration as follows:

$$\begin{aligned} \hat{X}_{k,n} &= \exp(\log Y_{k,n} - \frac{1}{N} \sum_{m=0}^{N-1} \log Y_{k,m}) \\ &= \frac{Y_{k,n}}{\exp(\frac{1}{N} \sum_{m=0}^{N-1} \log Y_{k,m})} = \frac{Y_{k,n}}{Q_k} \end{aligned} \tag{6}$$

where $\hat{X}_{k,n}$ is the equalized DFT energy, N is the number of frames in the sentence, and Q_k is the equalization factor for the kth DFT energy component in the current sentence. With this algorithm we can eliminate the stationary convolutional noise in the sentence without the assumption that H_l is constant within the spectral band.

B. Joint Channel and Model Estimation

Using CMN to perform channel equalization is tantamount to the underlying assumption that the sample cepstral average of the "clean" signal is an invariant quantity. This assumption is clearly violated when CMN is used to estimate the channel in short utterances. We present a different approach for jointly estimating the channel and the HMM parameters during training, and for obtaining the channel and the most likely state sequence during recognition.

In the cepstral domain, the observed speech signal corrupted by stationary convolutional noise can be written as

$$\mathbf{y}_n = \mathbf{h} + \mathbf{x}_n \tag{7}$$

where $\mathbf{h}$ is the cepstrum of the channel response, $\mathbf{x}_n$ is the clean speech cepstrum at each frame $n = 0, \ldots, N-1$ in the sentence, and we assume that the channel characteristics do not vary with time over a single sentence. In CMN the estimated channel $\hat{\mathbf{h}}$ is computed as a time average of all the frames in the sentence

$$\hat{\mathbf{h}} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{y}_n = \mathbf{h} + \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_n. \tag{8}$$

If we assume that the sequence $\mathbf{x}_n$ is modeled using HMM's with Gaussian observation distributions, then CMN will give an unbiased estimate of $\mathbf{h}$ only when $(1/N) \sum \mathbf{x}_n$ is zero, or more generally, independent of the sequence of distributions that generated $\mathbf{x}_n$.

In practice, the above average will not be constant since it depends on the sequence of distributions that generated $\mathbf{x}_n$, that is, on the transcription of the sentence. The CMN algorithm will interpret these fluctuations as channel variations, and remove them. In effect, this introduces an error in the true speech cepstrum, which may lead to recognition errors. A better approach is to try to jointly estimate the channel and the HMM parameters during training, and the channel and the state sequence during recognition.

Let us first assume that the HMM state sequence $[s_n], n = 0, \ldots, N-1$ is given. Then, the maximum-likelihood channel estimate is given by

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\operatorname{argmax}} \quad p(\mathbf{Y} \mid S, \theta, \mathbf{h}) \tag{9}$$

where $\mathbf{Y}$ is the collection of observations, S is the state sequence, θ are the HMM parameters, and $\mathbf{h}$ is the channel. For Gaussian output distributions, it can be shown [3] that this estimate is given by

$$\hat{\mathbf{h}} = \left[\sum_n (\mathbf{C}(s_n))^{-1} \right]^{-1} \sum_n (\mathbf{C}(s_n))^{-1} (\mathbf{y}_n - \mu(S_n)) \tag{10}$$

where the HMM output distribution

$$p(\mathbf{x}_n \mid s_n) = \mathcal{N}(\mu(s_n): \mathbf{C}(s_n)) \tag{11}$$

is a multivariate normal distribution with a state dependent mean $\mu(s_n)$ and covariance $\mathbf{C}(s_n)$. Hence, when the state HMM sequence is given, the channel estimate $\hat{\mathbf{h}}$ can be obtained as a weighted combination of the deviations of the observed features from the means of the HMM output distributions that are specified by that state sequence. The weights depend on the covariances of these output distributions. For HMM's with continuous mixtures as output distributions, (10) can be applied when both the state and the mixture index are known.

[1] The waveform is subdivided in a sequence of overlapping segments called frames, usally at intervals of 10–20 ms. Each frame is windowed before computing the DFT.

TABLE I
ERROR RATE AND DISTORTION FOR 18 *WSJO* DEVELOPMENT TEST SPEAKERS

Spkr Index	Senn Error Rate	OMic Error Rate	Error Ratio (OMic/ Senn)	Mic	Relative Distortion						
					Cep	D Cep	DD Cep	Egy	D Egy	DD Egy	Avg
426	7.3	5.2	0.7	A	0.60	0.57	0.60	0.23	0.19	0.20	0.40
22h	6.3	8.0	1.3	B	0.56	0.58	0.61	0.40	0.41	0.44	0.50
22k	12.5	16.8	1.3	B	0.48	0.54	0.58	0.34	0.27	0.30	0.42
052	9.0	10.4	1.2	C	0.65	0.66	0.68	0.73	0.55	0.57	0.64
061	8.2	11.0	1.3	C	0.59	0.62	0.65	0.65	0.50	0.53	0.59
00b	15.7	24.8	1.6	C	0.60	0.61	0.63	0.68	0.47	0.50	0.58
001	5.6	6.9	1.2	D	0.62	0.59	0.61	0.58	0.43	0.45	0.55
00d	21.0	34.5	1.6	D	0.72	0.73	0.77	0.49	0.31	0.32	0.56
22l	10.4	17.2	1.7	D	0.58	0.62	0.65	0.53	0.47	0.50	0.56
22g	6.7	11.9	1.8	D	0.62	0.68	0.72	0.60	0.51	0.54	0.61
431	17.7	32.5	1.8	E	0.63	0.65	0.67	0.70	0.50	0.51	0.61
422	20.9	40.1	1.9	F	0.92	0.81	0.82	0.38	0.31	0.33	0.60
400	13.8	30.7	2.2	G	0.83	0.81	0.83	0.53	0.61	0.65	0.71
423	9.6	24.8	2.6	G	1.00	0.87	0.87	0.43	0.50	0.55	0.70
424	12.3	32.0	2.6	G	0.99	0.90	0.92	0.52	0.63	0.68	0.77
00c	16.5	38.5	2.3	H	0.78	0.79	0.82	1.14	0.74	0.76	0.84
051	8.3	23.1	2.8	H	0.80	0.86	0.90	1.20	0.69	0.72	0.86
060	8.7	24.8	2.9	H	0.76	0.77	0.79	0.97	0.66	0.69	0.77
Avg	11.7	21.8	1.8		0.71	0.70	0.73	0.62	0.49	0.51	0.63

Below we examine how this channel estimate can be incorporated in the training and recognition problems.

Training: When the state sequence is not given, then one can use the expectation-maximization (EM) algorithm [4] to jointly estimate the channel and the HMM parameters by maximizing at each iteration the objective function

$$(\theta_N, \mathbf{h}_N) = \underset{\theta,\mathbf{h}}{\operatorname{argmax}} \quad E\{\log p(\mathbf{Y}, S \mid \theta, \mathbf{h}) \mid \mathbf{Y}, \theta_0, \mathbf{h}_0\} \tag{12}$$

where θ_0 and $\mathbf{h}_0$ are the parameters from the previous iteration, and θ_N and $\mathbf{h}_N$ are the reestimated parameters.

The solution to the maximization problem above is fairly complex, however, and the channel and model estimates can alternatively be obtained by an iterative procedure, where one alternates between obtaining estimates of the model parameters and the most likely state sequence, and using these estimates to compute the estimate for the stationary channel. Each iteration of the algorithm is therefore broken down into two steps:

1) Using the previous channel estimate $\mathbf{h}_0$, reestimate the model parameters using a nested EM procedure:

$$\theta_N = \underset{\theta}{\operatorname{argmax}} E\{\log p(\mathbf{Y}, S \mid \theta, \mathbf{h}_0) \mid \mathbf{Y}, \theta_0, \mathbf{h}_0\} \tag{13}$$

where S denotes the most likely state sequence using the current model and channel estimate.

2) Obtain a new channel estimate by maximizing the likelihood of the observations given the newly obtained model parameters θ_N

$$\mathbf{h}_N = \underset{\mathbf{h}}{\operatorname{argmax}} \quad p(\mathbf{Y} \mid \theta_N, \mathbf{h}). \tag{14}$$

The EM procedure described in (13) guarantees that the likelihood will not decrease for a fixed channel estimate, that is

$$\log p(\mathbf{Y} \mid \theta_N, \mathbf{h}_0) \geq \log p(\mathbf{Y} \mid \theta_0, \mathbf{h}_0). \tag{15}$$

For fixed HMM parameters, (14) also guarantees that the likelihood does not decrease

$$\log p(\mathbf{Y} \mid \theta_N, \mathbf{h}_N) \geq \log p(\mathbf{Y} \mid \theta_N, \mathbf{h}_0). \tag{16}$$

Therefore, every combined iteration of (13) and (14) guarantees that the likelihood $p(\mathbf{Y} \mid \theta, \mathbf{h})$ does not decrease. For simplicity, however, and if we assume that the most likely state sequence is dominant [5], we can replace (14) by

$$\mathbf{h}_N = \underset{\mathbf{h}}{\operatorname{argmax}}\, p(\mathbf{Y} \mid S, \theta_N, \mathbf{h}) \tag{17}$$

and the channel estimate above can be computed using (10).

Recognition: In recognition, we want to determine the most likely state sequence. This implies that we should jointly maximize over the state sequence and the channel

$$\max_{S,\mathbf{h}} \quad p(\mathbf{Y}, S \mid \theta, \mathbf{h}) \tag{18}$$

and the maximization above can be performed by an alternation between maximizing over the state sequence and over the channel estimate, which is similar to the training algorithm described in the previous section.

To summarize, we presented algorithms that jointly estimate the HMM parameters and the channel during training, and the most likely state sequence and the channel during recognition. During training, we assume that the training data can be split into blocks and that the channel characteristics do not

TABLE II
LISTING OF MICROPHONE TYPES IN DEVELOPMENT TEST SET

Mic Type	Microphone Description
A	Radio Shack Pro-Unidirectional Highball 33-984
B	Sony ECM-55
C	Sony ECM-50PS
D	Crown PCC-160 Phase-Coherent Table-Top
E	Shure SM91 Unidirectional Condenser
F	AT&T 720 Handset with Speech over Local Telephone Lines
G	AT&T 720 Speaker Phone with Speech over Local Telephone Lines
H	Crown PZM-6FS Pressure Zone Table-Top

vary with time within each block. These blocks can be either single utterances, or sessions with multiple utterances. A single estimate of the channel response in the cepstral domain is estimated for each block. The training algorithm alternates between estimating the channel response and using the new channel estimate to obtain refined estimates for the HMM parameters. Hence, the output distributions directly model the cepstrum of the clean signal. During recognition, an initial channel estimate based on *a priori* knowledge is used to obtain the most likely state sequence. This state sequence can then be used to refine the channel estimate using (10), and the procedure can be iterated.

C. Experimental Results

To compare both normalization algorithms presented in Section II–A and Section II–B to the conventional CMN algorithm, we tested the algorithms using SRI's DECIPHER™ continuous speech recognition system [6], [7] on the 5,000-word alternate microphone task of the *WSJ* corpus. The system is configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first-and second-order differences. The cepstral features are computed from an FFT filterbank. We used genonic hidden Markov models that allow an arbitrary degree of Gaussian sharing across different HMM states as described in [6]. For fast experimentation, we used the progressive search framework [7]: An initial recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using our channel normalization algorithms. The models were trained using the large-vocabulary *WSJ* corpus recorded with a close-talking Sennheiser[2] microphone from male speakers. For testing we used a test set with simultaneous recordings. One channel contains speech recorded with the Sennheiser microphone, and the other channel was recorded using 8 different low-quality microphones and telephone handsets. There were 18 male speakers in the test set. Each speaker recorded 20 sentences, for a total of 360 sentences. In Table I, the different speakers are grouped by secondary microphone type. The secondary microphone types are listed in Table II. We first compared the LDMN algorithm to the conventional CMN. In this experiment we used a tied-mixture HMM system, with all HMM states sharing the same mixture components.

TABLE III
WSJ 5K NVP DEVELOPMENT TEST SET WORD ERROR RATE

Algorithm	Sennheiser Microphone	Other Microphone
Cepstral Mean Removal	14.5	22.8
DFT Equalization	14.4	22.6

TABLE IV
CHANNEL EQUALIZATION RESULTS ON *WSJ* DEVELOPMENT TEST SET

Algorithm	Word Error Rate (%)
CMN	21.6
Channel Estimation	21.4

For each speaker, the word error rate[3] is given in Table I for the Sennheiser channel as well as the secondary microphone channel (denoted "OMic" for "other microphone"). The ratio of these word-error rates is shown in the fourth column. The normalized mean-squared error distortion between the Sennheiser and the secondary microphone features was computed for each of the six features. They are listed in subsequent columns, followed by an average of all six distortions. Note that the word-error rate and the average distortion are fairly constant across speakers for a given OMic condition. The results, presented in Table III, show that CMN is as effective as the LDMN equalization algorithm. To explain this result, we can either assume that the variation of convolutional noise within a spectral band is negligible, or that there are other factors that swamp its effects on recognition performance.

In a second experiment, we compared the joint channel and model estimation algorithm to CMN on the same database. The joint channel/model estimation algorithm was implemented as follows. At each iteration during training, the most likely

[2] All product names used in this paper are the trademark of their respective holders.

[3] The average word error rate in Table I is slightly different than the one shown in Table III because each has been computed with different training procedures.

TABLE V
INTER-SPEAKER VARIANCE OF THE CEPSTRAL MEAN MEASUREMENTS AS A PERCENTAGE OF THE TOTAL VARIANCE

c1	*c2*	*c3*	*c4*	*c5*	*c6*	*c7*	*c8*	*c9*	*c10*	*c11*	*c12*
71.4	79.8	60.1	70.8	49.5	71.4	53.5	78.0	68.0	51.1	25.1	57.9

state sequence was estimated for each utterance in the training set. Equalization was performed in the cepstral domain: A separate estimate of the channel response was obtained for each utterance using (10), and subsequently subtracted from the cepstral vectors. Compensation was followed by an iteration of the forward-backward algorithm. We computed a total of two iterations of the sequential EM algorithm during training. During recognition, an initial estimate of the channel was obtained using CMN. The most likely state sequence was obtained from the Viterbi alignment of a first recognition pass, and a more accurate estimate of the channel response was found using (10). A second recognition pass was then performed after subtracting the new channel estimate from the cepstral vectors.

The results are summarized in Table IV. In this experiment we used a phonetically-tied mixture system—that is, it had a smaller degree of mixture sharing than the tied-mixture system used in the first experiment. In this system, all context-dependent models with the same center phone use the same mixture components in their output distributions. Despite the serious channel mismatch between the Sennheiser recordings and the secondary-microphone recordings in the *WSJ* corpus, the results were essentially the same (21.6% with CMN and 21.4% with the proposed channel estimation algorithm). This indicates that the underlying assumption that $(1/N)\Sigma\mathbf{x}_n$ is independent of the sequence of distributions that generated $\mathbf{x}_n$ is fairly accurate for these long sentences (~8 seconds).

To test this hypothesis, we must compare for each speaker and channel the variation in the measurements of $(1/N)\Sigma\mathbf{x}_n$ when the transcription is fixed to the variation in the measurements of the same quantity when the transcription varies. To perform this comparison, we have to collect multiple recordings of each transcription for each speaker/channel combination. Assuming that the channel characteristics do not vary over the different recordings for a particular speaker/channel combination, we can then measure the cepstral mean for each sentence and group these measurements into sets based on the sentence transcription. Our hypothesis is then equivalent to the hypothesis that for each speaker/channel the averages of the cepstral mean values of the different groups are equal.

Since we did not have data to test this hypothesis directly, we measured the cepstral mean values for all 360 sentences in the test set. The variability in these measurements consists of two terms: the variability in the speaker/channel-dependent measurement of the channel h and the variability in the measurement of $(1/N)\Sigma\mathbf{x}_n$ (see (8)). Assuming that the channel characteristics do not vary during the 20-sentence section of each of the 18 speakers, then we can estimate each one of these two sources of variability by comparing the variance of the cepstral mean measurements within each 20-sentence section to the total variance. The results of these measurements for all 12 cepstral coefficients are presented in Table V, where we show the inter-speaker squared error as a percentage of the total squared error. We can see that the inter-speaker variance represents the larger amount of the total variance for most cepstral coefficients.

TABLE VI
PARAMETERS USED IN THE HIGH-QUALITY (HQ) AND TELEPHONE-QUALITY (TQ) FRONT ENDS

Parameter	HQ	TQ
Sampling Rate	16 kHz	8 kHz
Number of FFT Coefficients	256	128
Number of Cepstral Coefficients	12	8
Number of Filters	25	18
Total Bandwidth	100-6400 Hz	300-3000 Hz

This result agrees with our experimental finding that for the long *WSJ* sentences a satisfactory estimate of the channel can be obtained using CMN. Hence, we decided to perform an additional experiment to investigate the effect that the length of the interval used to obtain the channel estimate has on the accuracy of the estimate. As usual, we assumed that the channel does not vary within each speaker's 20-sentence section. Under this assumption, we can accurately estimate the channel response in the cepstral domain by computing the average of each cepstral coefficient over the whole 20-sentence section. We can then use this channel estimate to compute the average error in the less accurate channel estimates that are obtained using shorter intervals. In Fig. 1 we have plotted the error in the channel estimate as a percentage of the total variance of the corresponding cepstral coefficient and as a function of the estimation interval's length. The plots are averaged over all the intervals, sentences and speakers. We can see that, for an estimation interval of 8 seconds, the estimation error is small, and varies from 1.2% to 4.8% for different cepstral coefficients. The average estimation error over all cepstral coefficients for 8-second long intervals is 2.5%.

III. TRAINING ISSUES

A. Construction of Telephone-Bandwidth Acoustic Models

Our objective is to train an HMM recognizer for over the telephone (OTP) applications without collecting specific training data for each task. For example, we would like to use available large speech corpora recorded with high-quality (HQ) microphones instead of collecting data over the telephone network. Here we show that the variability in the acoustics of the telephone quality (TQ) recordings has little impact on performance as long as: 1) cepstral mean normalization is used

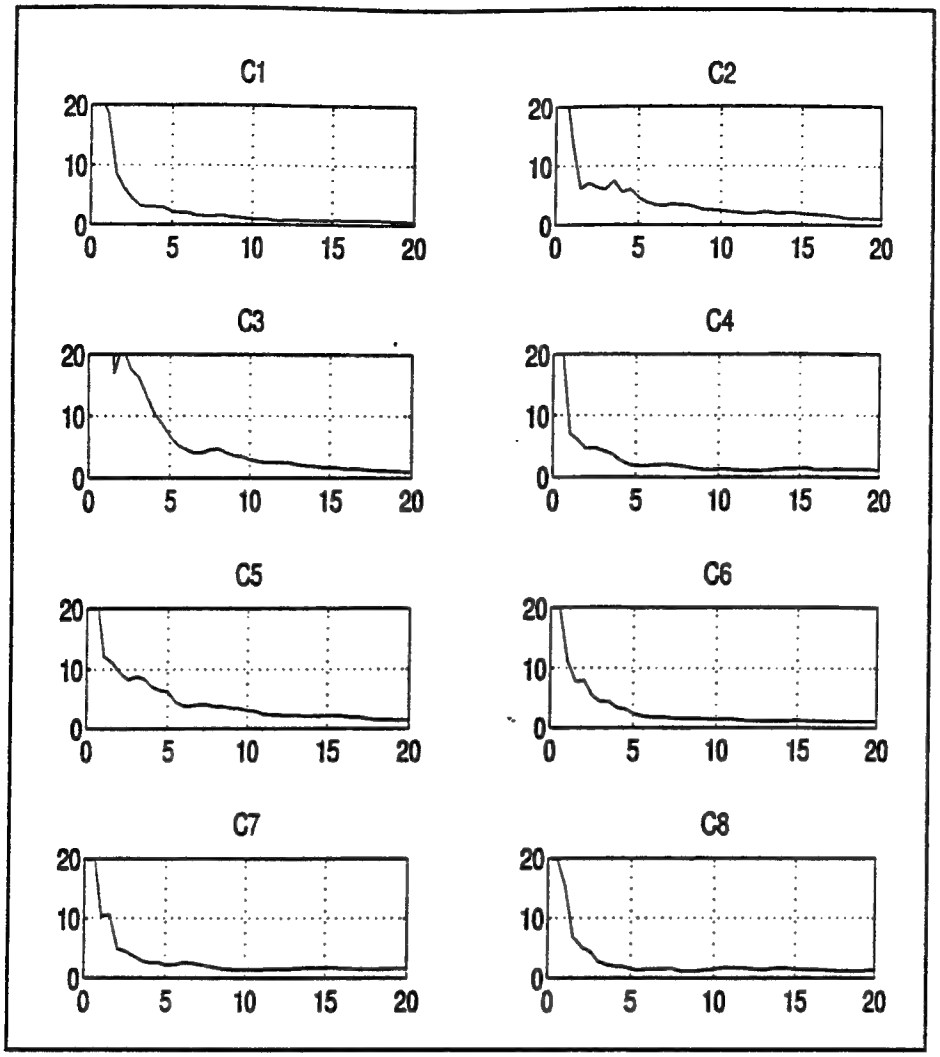

Fig. 1. Average error in the channel estimate (as a percentage of the total variance) as a function of the estimation interval (in seconds) for cepstral coefficients C1 through C8.

to compensate for channel variations, and 2) the signal analysis matches the spectrum of the telephone channel.

To avoid collecting new training data for a task in which there is a mismatch between training and test conditions, there are a number of possible approaches:

- Design robust features that are not affected by the variations in the microphone, background noise, channel distortion, and so forth.
- Adapt the parameters of the acoustic models.
- Map features between the test and train acoustic spaces. This means that we make the data used for testing look like the data used for training.

We will focus on the design of robust features for OTP applications by using a standard filterbank-based front end [8] tuned for telephone-bandwidth applications. In Table VI we show the parameters used in our wide-bandwidth (HQ) and telephone-bandwidth (TQ) front ends.[4] The main difference in the signal analysis stage is the total bandwidth of the filterbank. Both front-end signal processing modules produce six feature streams: cepstral energy (C0), cepstrum, and their first-and second-order differences. The mean of each cepstral coefficient is removed on a per-sentence basis.

[4] We shall use HQ/TQ to denote both the high-/telephone-quality data and the wide-/telephone-bandwidth front end, respectively.

B. Experimental Results on the ATIS Corpus

We have considered some of the approaches mentioned in Section III–A in the past [9], [10] and found that an adequate front end can minimize the mismatch between the acoustic spaces. In fact, in a pilot study conducted at SRI [9], we found that the variability introduced by the telephone handsets had little effect on recognition performance.

For our pilot study, we collected a corpus of both training and testing speech using simultaneous recordings made from subjects wearing a Sennheiser HMD 414 microphone and holding a telephone handset. The speech from the handset was sent over local telephone lines. Ten different handsets were used by 13 male subjects (10 for training and three for testing) who read ATIS (Air Travel Information System) sentences [11]. The selected telephones included three carbon button, two inexpensive Radio Shack, and a variety of telephones found in our lab. The amount of data was 3,000 sentences for training and 400 sentences for testing.

Table VII shows the results for different training and testing conditions. When the models are trained with HQ data and

TABLE VII
EFFECT OF DIFFERENT TRAINING AND SIGNAL PROCESSING ON TEST SET PERFORMANCE

Acoustic Model Training		Test Set Word Error (%)	
Training Data	Signal Processing	Sennheiser (HQ data)	Telephone (TQ data)
Sennheiser (HQ)	High-Quality Front End	7.8	19.4
Sennheiser (HQ)	Telephone Front End	9.0	9.7
Telephone (TQ)	Telephone Front End	10.0	10.3
Sennheiser (HQ)	Telephone Front End without Cepstral-Mean Normalization	9.4	11.2

the HQ front end is used to generate the features we get the best possible result in the train HQ/test HQ condition (7.8% word error rate) and the worst result when we test on the TQ data (19.4%). This shows how the error is doubled due to the mismatch in the higher frequencies of the spectrum. The difference in error rate between the test HQ and test TQ conditions is greatly reduced when the TQ front end is used (9.0% and 9.7% error, respectively). Here the robustness of the recognizer is increased at the expense of performance in the HQ test condition. The next line in the table shows that training the models with TQ data actually degrades performance even for the TQ test condition (10.0% and 10.3% for HQ test and TQ test conditions). This is an important result since it indicates that we can train TQ models using HQ data with no degradation in performance. This is no longer true when we eliminate the cepstral-mean normalization (CMN) algorithm [2], as shown in the last line of the table. This degradation in performance is caused by the stationary convolutional noise (9.4% and 11.2% for HQ test and TQ test conditions when CMN is not used).

In summary, we can train the recognizer models using a telephone bandwidth front end and high-quality training data. The drawback of the method, however, is that separate models have to be trained for HQ and TQ applications. Another limitation of this experiment is that all the telephone data were recorded using the same local telephone line. Therefore, we cannot predict from these experiments on a small stereo speech corpus how the variability of a wider telephone network will affect the recognition performance. For this reason, we test telephone models trained with HQ data on a more realistic database: the *Switchboard* speech corpus.

C. Experimental Results on the Switchboard Corpus

In this experiment we also show how to train HMM models for OTP applications using a HQ database and how they compare to models trained with TQ data. The test is performed on the *Credit-Card (CC)* task that is part of the *Switchboard* [12] speech corpus, a large speech database recorded over the public telephone network. For training we use the *WSJ* database that was recorded using high-quality Sennheiser microphones. The *CC* corpus consists of spontaneous telephone conversations between two individuals talking about issues related to credit cards. In contrast, the *WSJ* corpus was recorded from subjects reading sentences extracted from the *Wall Street Journal* newspaper.

To test our ideas on the *CC* task we decided to train the acoustic models using 7000 *WSJ* sentences. For the *CC* task, training the models with *WSJ* data presents mismatches along a number of dimensions, which include:

- Acoustics of recording (high-quality versus telephone)
- Vocabulary independence (*WSJ* does not have the same focus as the credit card conversations)
- Amount of training data (*WSJ* has 7000 training sentences, *CC* has 1000)
- Speaking modes (read versus spontaneous speech)

We ran the recognition experiments using SRI's DECIPHER™ phonetically-tied mixture system with a TQ front end. All the recognition experiments are gender-dependent, use a bigram grammar, and are expressed in terms of word error rate. The test consisted of 167 sentences. The results are summarized in Table VIII. In the baseline experiment, where we trained and tested the models using *CC* data, the error rate was 68%. The cross-database experiment yielded a slightly higher error of 71.5%. We also tested the *WSJ*-trained models with a noisy version of the test set (*n*CC). The data was corrupted with mid-continental US voice channel effects and highway noise recorded in the interior of a Ford Taurus on the highway. The average signal-to-noise ratio after adding the noise was 20 dB. The error for the *n*CC test set was 78.9%.

To improve performance in the cross-database experiment, we adapted the distributions of the HMM using the *CC* train set. To adapt the models we reestimated the parameters of the Gaussian distributions (means and variances) using the forward-backward algorithm [13]. The mixture weights and state transition probabilities remained unchanged. This approach reduced the error to 69.7%. Finally, we ran two additional iterations of the forward-backward algorithm on the *WSJ*-trained models using the *CC* train set. This run produced the best result of 67.1% error rate.

TABLE VIII
SUMMARY OF CROSS-DATABASE ACOUSTIC TRAINING RESULTS ON THE *CREDIT CARD* TASK

Description of the Experiment	Train Data	Test Data	Word Error (%)
Baseline	CC	CC	68.1
Cross-Database	WSJ	CC	71.5
Cross-Database in Noisy Data	WSJ	nCC	78.9
Adaptation of WSJ Gaussian Mixtures	WSJ/CC	CC	69.7
CC Booted from WSJ Models	WSJ/CC	CC	67.1

The cross-database results are very close to the baseline despite the mismatches between the two databases. Based on previous experiments, we believe that the difference in the results is more likely to be caused by mismatches in speaking modes and vocabulary than in the acoustics of the recording environment.

IV. SUMMARY

To compensate for channel and microphone mismatch we investigated the validity of two simplifying assumptions of the popular cepstral-mean normalization algorithm. To remove these assumptions, we introduced two new channel normalization algorithms. Our experimental results showed that on the *WSJ* alternate-microphone task the cepstral-mean normalization algorithm was as effective as the proposed channel normalization algorithms.

We also presented our approach to developing acoustic models for telephone applications. We showed that we can take advantage of existing, "high-quality" data and achieve similar performance with cross-database training to that obtained using task-dependent training.

REFERENCES

[1] G. Doddington, "CSR corpus development," in *DARPA SLS Workshop*, Feb 1992.

[2] S. F. Furui, "Cepstral analysis technique for automatic speaker verification," in *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-29, pp. 254–272, Apr. 1981.

[3] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood estimation from incomplete data," *J. Royal Stat. Soc.* (B), vol. 39, no. 1, pp. 1–38, 1977.

[5] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A segmental K—means training procedure for connected word recognition," *AT&T Tech. J.*, pp. 21–40. May-June 1986.

[6] V. Digalakis and H. Murveit, "Genones: Optimizing the degree of mixture-tying in a large-vocabulary HMM-based speech recognizer," in *Proc. ICASSP*, Apr. 1994, pp. I-537–I-540.

[7] H. Murveit *et al.*, "Large vocabulary dictation using SRI's DECIPHER™ speech recognition system: Progressive search techniques," in *Proc. ICASSP*, Apr. 1993, pp. II-319–II-322.

[8] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357–366, Aug. 1980.

[9] M. Weintraub and L. Neumeyer, "Constructing telephone acoustic models from a high quality speech corpus," *Proc. ICASSP*, Apr. 1994, pp. I-85–I-88.

[10] L. Neumeyer and M. Weintraub, "Probabilistic optimum filtering for robust speech recognition," in *Proc. ICASSP*, Apr. 1994, pp. I-417–I-420.

[11] MADCOW, "Multi-site data collection for a spoken language corpus," in *Proc. 1992 DARPA Speech, Natural Lang. Workshop*, pp. 7–14.

[12] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. IEEE ICASSP-92*, pp. I-517–I-520.

[13] L. E. Baum *et al.*, "A maximization technique in the statistical analysis of probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, vol. 41, pp. 164–171, 1970.

Leonardo G. Neumeyer (S'88–M'90) was born in Buenos Aires, Argentina. He received the Engineer degree in electronics from the University of Buenos Aires, in 1988, and the M.Eng. degree from Carleton University, Ottawa, Canada, in 1990.

From 1991 to 1992, he worked on the development of low bit rate speech coding systems at Novatel Communications in Calgary, Canada. In March 1992, he joined the Speech Technology and Research Laboratory at SRI International in Menlo Park, CA. His interests include digital signal processing, speech recognition, and speech coding.

Vassilios V. Digalakis was born in Hania, Greece, on February 2, 1963. He received the Diploma in electrical engineering from the National Technical University of Athens, Greece, in 1986, the M.S. degree in electrical engineering from Northeastern University, Boston, MA, in 1988, and the Ph.D. degree in electrical and systems engineering from Boston University, Boston, MA, in 1992.

From 1986 to 1988, he was a Teaching and Research Assistant at Northeastern University. From 1988 to 1991, he served as a Research Assistant at Boston University. In January 1992, he joined SRI International in Menlo Park, CA, where he is working on speech recognition. His research interests are in statistical signal processing, pattern recognition, and estimation theory, with applications to speech recognition, biomedical image processing, and digital communications.

Mitchel Weintraub received the B.S. degree in applied and engineering physics from Cornell University, Ithaca, NY, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1982 and 1985, respectively.

Since 1985, he has been with the Speech Technology and Research Laboratory at SRI International, Menlo Park, CA. His current research interests include robust speech recognition, spoken language systems, computational models of the auditory system, signal processing, and speech interference suppression.

SPEAKER ADAPTATION USING COMBINED TRANSFORMATION AND BAYESIAN METHODS

Vassilios Digalakis and Leonardo Neumeyer

SRI International
Speech Technology and Research Laboratory
Menlo Park, CA, 94025, USA

ABSTRACT

The performance and robustness of a speech recognition system can be improved by adapting the speech models to the speaker, the channel and the task. In continuous mixture-density hidden Markov models the number of component densities is typically very large, and it may not be feasible to acquire a large amount of adaptation data for robust maximum-likelihood estimates. To solve this problem, we propose a constrained estimation technique for Gaussian mixture densities, and combine it with Bayesian techniques to improve its asymptotic properties. We evaluate our algorithms on the large-vocabulary Wall Street Journal corpus for nonnative speakers of American English. The recognition error rate is comparable to the speaker-independent accuracy achieved for native speakers.

1. INTRODUCTION

Two families of adaptation schemes have been proposed in the past. One transforms the speaker's feature space to "match" the space of the training population [1],[2],[3]. The transformation can be applied either directly to the features, or to the speech models [4]. This approach has the advantage of simplicity and, if the number of free parameters is small, then transformation techniques adapt to the user with only a small amount of adaptation speech (quick adaptation). Disadvantages of transformation methods are that they are usually text-dependent and that they may not take full advantage of large amounts of adaptation data. The second main family of adaptation algorithms follows a Bayesian approach, where the speaker-independent information is encapsulated in the prior distributions [5][6]. The Bayesian approach is text-independent, and has nice asymptotic properties: speaker-adaptive performance will converge to speaker-dependent performance as the amount of adaptation speech increases. However, the adaptation rate is usually slow.

In this paper we present adaptation schemes that combine the quick adaptation characteristics of transformation-based methods with the nice asymptotic properties of Bayesian methods. We first introduce a transformation-based method for continuous mixture-density hidden Markov models (HMMs). Adaptation is achieved via a transformation of the speaker-independent observation densities, and the transformation parameters are obtained using the maximum-likelihood (ML) criterion. The number of transformation parameters can be adjusted to achieve quick adaptation. We will then show how this algorithm can be combined with Bayesian techniques. The combined method adapts to a new speaker with small amounts of adaptation data, but also has nice asymptotic properties and takes full advantage of large amounts of adaptation data.

2. TRANFORMATION-BASED ADAPTATION

Transformation-based approaches to speaker adaptation are typically text-dependent and require the new speaker to record some predetermined sentences. These utterances are aligned to ones recorded by reference speakers, and mappings between the new-speaker and the reference-speaker acoustic spaces are obtained using regression techniques [2][3].

We have developed a novel transformation-based approach to speaker adaptation for continuous mixture-density HMMs [7]. We apply the transformation at the distribution level, instead of transforming the feature vectors directly, since we can then use the expectation-maximization (EM) algorithm [8] to estimate the transformation parameters by maximizing the likelihood of the adaptation data. Using this approach, we are not required to time-align the new- and reference-speaker data, and the transformation parameters can be estimated using new-speaker data alone. Our scheme can also be viewed as a constrained estimation of Gaussian mixtures, since we apply the same transformation to all the components of a particular mixture (or a group of mixtures, if there is tying of transformations) instead of independently reestimating them. It achieves quick adaptation by adapting Gaussians for which there were no observations in the training data, based on data that were most likely generated by other Gaussians of the same or neighboring mixtures.

Specifically, we assume that the speaker-independent (SI) HMM model for the SI vector process $[y_t]$ has observation densities of the form

$$p_{SI}(y_t \mid s_t) = \sum_i p(\omega_i \mid s_t) N(y_t; \mu_{ig}, \Sigma_{ig}) \quad , \quad (1)$$

where g is the index of the Gaussian codebook used by state s_t. Adaptation of this system can be achieved by jointly transforming all the Gaussians of each mixture. We assume that, given the HMM state index s_t, the speaker-dependent vector process $[x_t]$ can be obtained by an underlying process $[y_t]$ through the transformation

$$x_t = A_g y_t + b_g \quad . \tag{2}$$

Under this assumption, the speaker-adapted (SA) observation densities will have the form

$$p_{SA}(x_t|s_t) = \sum_i p(\omega_i|s_t)\, N\left(x_t; A_g\mu_{ig} + b_g, A_g\Sigma_{ig}A_g^T\right) \tag{3}$$

and only the parameters $A_g, b_g, g = 1, \ldots, N_g$ need to be estimated during adaptation, where N_g is the number of distinct transformations. The same transformations can be applied to different HMM states, and this tying of transformations can be used to optimize performance based on the amount of available adaptation data. The transformation parameters can be estimated using the EM algorithm. The reestimation formulae are derived in [7] and are summarized below:

1. Initialize all transformations with $A_g(0) = I, b_g(0) = 0, g = 1, \ldots, N$. Set $k=0$.
2. **E-step:** Perform one iteration of the forward-backward algorithm on the speech data, using Gaussians transformed with the current value of the transformations $A_g(k), b_g(k)$. For all component Gaussians and all mixtures g, collect the sufficient statistics

$$\begin{aligned}
\mu_{ig} &= \frac{1}{n_{ig}}\sum_{t,s_t}\gamma_t(s_t)\,\phi_{it}(s_t)\,x_t \\
\Sigma_{ig} &= \frac{1}{n_{ig}}\sum_{t,s_t}\gamma_t(s_t)\,\phi_{it}(s_t)\,(x_t-\mu_{ig})(x_t-\mu_{ig})^T \\
n_{ig} &= \sum_{t,s_t}\gamma_t(s_t)\,\phi_{it}(s_t)
\end{aligned} \tag{4}$$

where $\gamma_t(s_t)$ is the probability of being at state s_t at time t given the current HMM parameters, the summation is over all times and HMM states that share the same mixture components, and $\phi_{it}(s_t)$ is the posterior probability

$$\phi_{it}(s_t) = p(\omega_{ig} \mid A_g(k), b_g(k), x_t, s_t) \quad . \tag{5}$$

3. **M-step:** Compute the new transformation parameters. Under the assumption of diagonal covariance and transformation matrices, the elements a and b of $A_g(k+1), b_g(k+1)$ can be obtained by solving the following equations for each g

$$\begin{aligned}
b &= \left(\sum_i \frac{n_i\mu_i}{\sigma_i^2} - a\sum_i \frac{n_i\mu_i}{\sigma_i^2}\right)\Big/\left(\sum_i \frac{n_i}{\sigma_i^2}\right) \\
&\left(\sum_i n_i\right)a^2 - \left(\sum_i \frac{n_i}{\sigma_i^2}\right)b^2 - \left(\sum_i \frac{n_i\mu_i}{\sigma_i^2}\right)ab \\
&+ \left(\sum_i \frac{n_i\mu_i\mu_i}{\sigma_i^2}\right)a + \left(2\sum_i \frac{n_i\mu_i}{\sigma_i^2}\right)b - \left(\sum_i n_i\frac{\mu_i^2+\sigma_i^2}{\sigma_i^2}\right) = 0
\end{aligned} \tag{6}$$

where for simplicity we have dropped the dependence on g. The variables $\mu_i, \sigma_i^2, \mu_i, \sigma_i^2$ are elements of the vectors and diagonal matrices $\mu_{ig}, \Sigma_{ig}, \mu_{ig}, \Sigma_{ig}$, respectively.

4. If the convergence criterion is not met, go to step 2.

Once the transformation parameters are determined, the constrained ML estimates for the means and covariances can be obtained using

$$\begin{aligned}
\mu_{ig}^{CML} &= A_g\,\mu_{ig} + b_g \\
\Sigma_{ig}^{CML} &= A_g\Sigma_{ig}A_g^T
\end{aligned} \quad . \tag{7}$$

3. COMBINED TRANSFORMATION AND BAYESIAN-BASED ADAPTATION

In Bayesian adaptation techniques the limited amount of adaptation data is optimally combined with the prior knowledge. With the appropriate choice of the prior distributions, the maximum *a posteriori* (MAP) estimates for the means and covariances of HMMs with Gaussian observation densities can be obtained using linear combinations of the speaker-dependent sufficient statistics (counts) and some quantities that depend on the parameters of the prior distributions [5][6]. Based on the reestimation formulae for the MAP estimates of the means and covariances of HMM with continuous mixture densities that are derived in [6], a simplified version of Bayesian estimation can be implemented by linearly combining the speaker-independent and the speaker-dependent counts for each component density

$$\begin{aligned}
\langle x\rangle_{ig}^{SA} &= \lambda\langle x\rangle_{ig}^{SI} + (1-\lambda)\,\langle x\rangle_{ig}^{SD} \\
\langle xx^T\rangle_{ig}^{SA} &= \lambda\langle xx^T\rangle_{ig}^{SI} + (1-\lambda)\,\langle xx^T\rangle_{ig}^{SD} \\
n_{ig}^{SA} &= \lambda n_{ig}^{SI} + (1-\lambda)\,n_{ig}^{SD}
\end{aligned} \quad , \tag{8}$$

where the superscripts denote the data over which the following statistics are collected during one iteration of the forward-backward algorithm

$$\begin{aligned}
\langle x\rangle_{ig} &= \sum_{t,s}\gamma_t(s)\,\phi_{it}(s)\,x_t \\
\langle xx^T\rangle_{ig} &= \sum_{t,s}\gamma_t(s)\,\phi_{it}(s)\,x_t x_t^T \\
n_{ig} &= \sum_{t,s}\gamma_t(s)\,\phi_{it}(s)
\end{aligned} \quad . \tag{9}$$

We will refer to this method as approximate Bayesian adaptation. The weight λ controls the adaptation rate. Using the combined counts, we can compute the approximate MAP (AMAP) estimates of the means and covariances of each Gaussian component density from

$$\begin{aligned}
\mu_{ig}^{AMAP} &= \frac{\langle x\rangle_{ig}^{SA}}{n_{ig}^{SA}} \\
\Sigma_{ig}^{AMAP} &= \frac{\langle xx^T\rangle_{ig}^{SA}}{n_{ig}^{SA}} - \mu_{ig}^{AMAP}\left(\mu_{ig}^{AMAP}\right)^T
\end{aligned} \quad . \tag{10}$$

Similar adaptation schemes have also appeared for discrete HMMs [9], and can be used to adapt the mixture weights in the approximate Bayesian scheme described here.

In Bayesian adaptation schemes, only the Gaussians of the speaker-independent models that are most likely to have generated some of the adaptation data will be adapted to the speaker. These Gaussians may represent only a small fraction of the total number in continuous HMMs with a large number of Gaussians. On the other hand, as the amount of adaptation data increases, the speaker-dependent statistics will dominate the speaker-independent priors and Bayesian techniques will approach speaker-dependent performance. We should, therefore, aim for an adaptation scheme that retains the nice properties of Bayesian schemes for large amounts of adaptation data, and has improved performance for small amounts of adaptation data. We can achieve this by using our transformation-based adaptation as a preprocessing step to transform the speaker-independent models so that they better match the new speaker characteristics and improve the prior information in MAP estimation schemes. In the approximate Bayesian adaptation, this can be accomplished by first transforming the speaker-independent counts using the method described in Section 2 and then combining them with the speaker-dependent counts collected using the adaptation data.

4. EXPERIMENTAL RESULTS

We evaluated our adaptation algorithms on the "spoke 3" task of the phase-1, large-vocabulary Wall Street Journal (WSJ) corpus [10], trying to improve recognition performance for non-native speakers of American English. Experiments were carried out using SRI's DECIPHER™ speech recognition system configured with a six-feature front end that outputs 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. The cepstral features are computed from a fast Fourier transform (FFT) filterbank, and subsequent cepstral-mean normalization on a sentence basis is performed. We used genonic hidden Markov models with an arbitrary degree of Gaussian sharing across different HMM states as described in [11]. The speaker-independent continuous HMM systems that we used as seed models for adaptation were gender-dependent, trained on 140 speakers and 17,000 sentences for each gender. Each of the two systems had 12,000 context-dependent phonetic models that shared 500 Gaussian codebooks with 32 Gaussian components per codebook. For fast experimentation, we used the progressive search framework [12]: an initial, speaker-independent recognizer with a bigram language model outputs word lattices for all the utterances in the test set. These word lattices are then rescored using speaker-adapted models. We used the baseline 5,000-word, closed-vocabulary bigram and trigram language models provided by the MIT Lincoln Laboratory. The trigram language model was implemented using the N-best rescoring paradigm, by rescoring the list of the N-best sentence hypotheses generated using the bigram language model.

In the first series of experiments we used the bigram language model. We first evaluated the performance of the transformation-based adaptation for various numbers of transformations and amounts of adaptation data. As we can see in Figure 1,

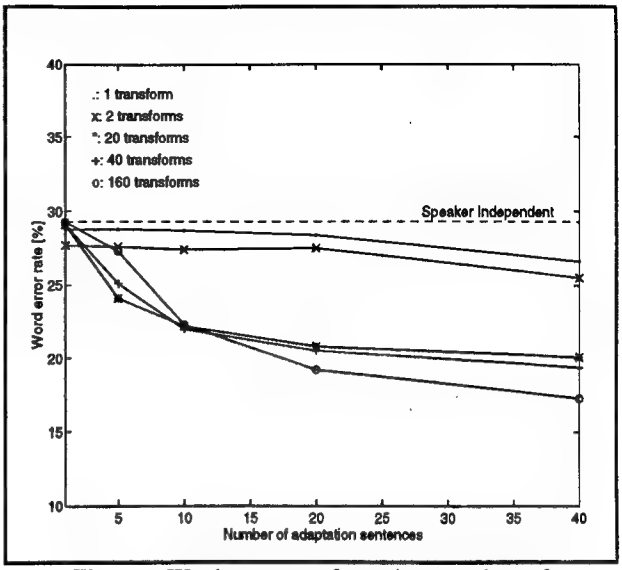

Figure 1: Word error rates for various numbers of transformations for the transformation-based adaptation

where we have plotted the word error rate as a function of the number of adaptation sentences, multiple transformations outperform very constrained schemes that use 1 or 2 transformations. The performance with 20 and 40 transformations is similar, and is better than the less constrained case of 160 transformations. However, as the amount of adaptation data increases, the 160 transformations take advantage of the additional data and outperform the more constrained schemes. A significant decrease in error rate is obtained with as few as 5 adaptation sentences. When adapting using a single sentence, the performance is similar for different numbers of transformations, except for the case of 2 transformations. The reason is that in our implementation a transformation is reestimated only if the number of observations is larger than a threshold; otherwise, we use a global transformation estimated from all data. Since most of the transformations are backed off to the global transformation for the case of a single adaptation sentence, the cases with different numbers of transformations exhibit similar performance.

In Figure 2 we compare the word error rates of the transformation-only method with 20 and 160 transformations, the approximate Bayesian method with conventional priors, and the combined method for various amounts of adaptation data. In the latter, the number of transformations was optimized according to the available amount of adaptation data. The transformation-only method with 20 transformations outperforms the Bayesian scheme with conventional priors when fewer than 10 sentences are used for adaptation, whereas the situation reverses as more adaptation sentences are used. This is consistent with our claim that transformation-based methods adapt faster, whereas Bayesian schemes have better asymptotic properties. The performance of the transformation approach for large amounts of adaptation data can be improved by increasing the number of transformations. We can also see in the same figure the success of the combined method, which significantly outperforms the first two methods over the whole range of adaptation sentences

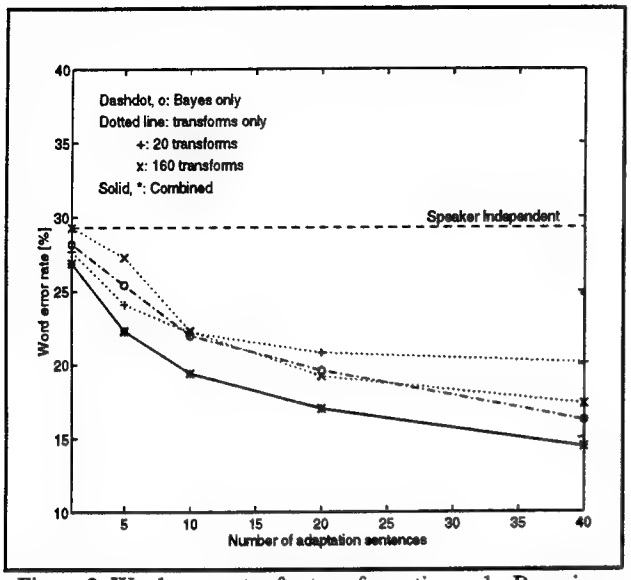

Figure 2: Word error rates for transformation-only, Bayesian-only, and combined schemes.

that we examined. The transformation step provides quick adaptation when few adaptation sentences are used, and the Bayesian reestimation step improves the asymptotic performance.

Finally, we evaluated the word error rate of our best-performing configuration on the 1993 Spoke-3 development and evaluation sets, and the 1994 evaluation set of the WSJ corpus using a trigram language model. Our results for the 1993 test sets, presented in Table 1, represent the best reported results to date on this task [13][1]. The speaker-independent word error rate for nonnative speakers is reduced by a factor of 2 using only 40 adaptation sentences. Using 200 adaptation sentences, the speaker-adapted error rate of nonnative speakers is comparable to the native speaker-independent word error rate of the same recognition system which is 7.2% and 8.1% on the 1993 development and 1994 evaluation sets, respectively.

Test Set	Adaptation Sentences	SI rate (%)	SA rate (%)
Dev. 93	40	23.5	10.3
Eval. 93	40	16.5	10.0
Eval. 94	40	23.2	11.3
	100		9.4
	200		8.2

Table 1. Speaker Independent (SI) and Speaker Adapted (SA) word error rates on various test sets of nonnative speakers using different amounts of adaptation data.

1. The 1994 official ARPA benchmark results were not available when this paper was written.

Acknowledgments

We gratefully acknowledge support for this work from ARPA through Office of Naval Research Contracts N00014-93-C-0142 and N00014-92-C-0154. The Government has certain rights in this material. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Government funding agencies.

REFERENCES

1. J. Bellegarda *et al.*, "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping," 1992 IEEE ICASSP, pp. I-445—I-448.
2. K. Choukri, G. Chollet and Y. Grenier, "Spectral Transformations through Canonical Correlation Analysis for Speaker Adaptation in ASR," 1986 IEEE ICASSP, pp. 2659—2662.
3. S. Nakamura and K. Shikano, "A Comparative Study of Spectral Mapping for Speaker Adaptation," 1990 IEEE ICASSP, pp. 157—160.
4. A. Sankar and C.-H. Lee, "Stochastic Matching for Robust Speech Recognition," *IEEE Signal Processing Letters,* Vol.1, No.8, August 1994.
5. C.-H. Lee, C.-H. Lin and B.-H. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models," *IEEE Trans. on Acoust., Speech and Signal Proc.,* Vol. ASSP-39(4), pp. 806—814, April 1991.
6. C.-H. Lee and J.-L. Gauvain, "Speaker Adaptation Based on MAP Estimation of HMM Parameters," 1993 IEEE ICASSP, pp. II-558 — II-561.
7. V. Digalakis, D. Rtischev and L. Neumeyer, "Fast Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," submitted to *IEEE Trans. on Speech and Audio Processing,* April 1994.
8. A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood Estimation from Incomplete Data," *Journal of the Royal Statistical Society (B)*, Vol. 39, No. 1, pp. 1—38, 1977.
9. X. Huang and K.-F. Lee, "On Speaker-Independent, Speaker-Dependent and Speaker-Adaptive Speech Recognition," *IEEE Trans. on Speech and Audio Processing,* Vol. 1, No. 2, pp. 150—157, April 1993.
10. D. Paul and J. Baker, "The Design for the Wall Street Journal-based CSR corpus," 1992 DARPA Speech and Natural Language Workshop, pp. 357—362.
11. V. Digalakis and H. Murveit, "Genones: Optimizing the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," 1994 IEEE ICASSP, pp. I-537—I-540.
12. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques," 1993 IEEE ICASSP, pp. II-319—II-322.
13. D. Pallet *et al.*, "1993 Benchmark Tests for the ARPA Spoken Language Program," 1994 ARPA HLT Workshop.

ROBUST SPEECH RECOGNITION IN NOISE USING ADAPTATION AND MAPPING TECHNIQUES

Leonardo Neumeyer and Mitchel Weintraub

SRI International
Speech Technology and Research Laboratory
Menlo Park, CA, 94025, USA

ABSTRACT

This paper compares three techniques for recognizing continuous speech in the presence of additive car noise: 1) transforming the noisy acoustic features using a mapping algorithm, 2) adaptation of the Hidden Markov Models (HMMs), and 3) combination of mapping and adaptation. We show that at low signal-to-noise ratio (SNR) levels, compensating in the feature and model domains yields similar performance. We also show that adapting the HMMs with the mapped features produces the best performance. The algorithms were implemented using SRI's DECIPHER™ speech recognition system [1-3] and were tested on the 1994 ARPA-sponsored CSR evaluation test spoke 10 [4].

1. INTRODUCTION

There are several approaches that one can use to recognize speech in the presence of additive background noise. The algorithms that we present here attempt to make each of the major components robust to additive noise: (a) the front-end signal processing and (b) the statistical modeling.

To make the signal processing robust to additive noise, we apply a technique called *Probabilistic Optimum Filtering* (POF) [5]. We have previously showed how this algorithm can be used to recognize narrowband speech recorded over the telephone using wideband HMMs, and how to map speech features obtained from a boom desktop microphone to features generated from a close talking microphone. In summary, our focus in developing POF was the problem of channel mismatches between training and testing conditions.

The class of feature-transformation approaches have been used successfully by other researchers [6,7] to compensate for speech corrupted with additive noise. We extend these techniques by using the POF technique and combine it with the ideas in our earlier noise-robust work [8]. Specifically, we train many different POF filters for different conditions (e.g. different background noise, different SNR levels). At runtime, we automatically select the most appropriate model.

The POF model does not use any assumption about the underlying physical phenomena that corrupted the signal. However, it requires stereo recordings of the clean and noisy speech to estimate its parameters. In the case of additive noise, it is straightforward to build an artificial stereo database when a sample of the noise is available, just by adding the noise to the clean speech.

One approach to make the statistical modeling robust to additive noise is Parallel Model Combination (PMC) [9]. PMC is used to adapt the HMM parameters in a very simple but effective manner and it has also been shown [10] that integrating PMC with a continuous spectral subtraction in the front end is beneficial at low SNRs.

Our approach to robust statistical modeling is to use a model adaptation technique described in [11]. In this case, we apply a set of affine transformations to the Gaussian mixtures of the HMMs. Unlike POF, stereo data are not needed to estimate the adaptation parameters. The clean HMMs are adapted using an orthographically transcribed adaptation set that matches the noisy conditions.

Finally, we investigate how both techniques (mapping and adaptation) perform when they are used together. That is, we enhance the noisy features using POF followed by the adaptation stage. In fact, at low SNRs this technique produces the best performance.

2. COMPENSATION TECHNIQUES

2.1. Feature Mapping

The POF mapping algorithm is designed to estimate a clean feature vector by applying a set of weighted affine transformations to the noisy feature vectors [5]. To estimate the POF transformation parameters, we need a stereo compensation set with simultaneous sequences of the clean and noisy feature vectors. The stereo data is created by adding noise to the clean data to obtain noisy data. The question arises as to what noise to add to the clean speech and how the transformation parameters are affected by the properties of the noise (spectrum and level). Three possible approaches are to (1) add many different types of noise to the training data and train a general mapping that will apply to all types of additive noise, (2) train many different mappings for different noise spectra and SNR's, and select the best model at runtime, and (3) obtain a sample of the actual noise encountered in the field and build a specific mapping for these conditions at runtime.

2.2. Model Adaptation

In the feature-mapping approach clean features are estimated and the HMMs remain unchanged. In model adaptation, however, the opposite occurs: the noisy feature vectors are unchanged and the HMMs are adapted using a sample of the noisy speech data and its orthographic transcription.

Adaptation of the HMMs is implemented using a constrained estimation of the Gaussian mixtures [11]. In this algorithm, we estimate a set of affine transformations that are applied to the Gaussian distributions. The transformations can be either unique for each mixture of Gaussians or shared by different mixtures. The total number of transformations is determined experimentally based on the amount of adaptation data.

As in the mapping approach, the compensation set can be constructed using a variety of speakers and noises. To achieve good performance, however, the characteristics of the noise and the SNR in the adaptation set have to match the test conditions.

2.3. Combination of Mapping and Adaptation

The third approach adapts the HMMs using the mapped feature vectors. In this algorithm, the feature mapping transforms the noisy features to make them look like the clean features. Then, the HMMs are adapted to these mapped noisy features. Finally, at runtime, the POF mapping is applied to the noisy features and these features are recognized with the adapted HMMs.

This approach might be particularly applicable at low SNRs where the mapped features may be significantly distorted, and the adaptation algorithm is not able to compensate the models in the cepstral domain because of the highly nonlinear distortion introduced by the additive noise.

3. EXPERIMENTS

Section 3.1 compares the POF, the HMM adaptation, and the combined approach for various SNR levels. Section 3.2 summarizes the procedure used for the 1994 ARPA-sponsored benchmark tests on noisy channels.

3.1. Comparison of Compensation Techniques

We evaluated the noise compensation algorithms on the large vocabulary Wall Street Journal (WSJ) corpus [12]. The experiments were carried out using SRI's DECIPHER™ speech recognition system [1-3] configured with a six-feature front end: 12 cepstral coefficients, cepstral energy, and their first- and second-order differences. We used genonic HMMs, as described in [1]; for rapid experimentation, we constrained the search using the Progressive Search Technique described in [2]. In the current section (Section 3.1) we used lattices created on the clean test set (before adding the noise) to constraint the recognition search, resulting in optimistic results. In the following section (Section 3.2), we use a full search decoder, resulting in real error rates.

The noisy data were created artificially in the lab by adding the scaled noise to the speech data. Eight minutes of car noise were recorded on a 1985 Honda Civic Station Wagon traveling at a steady speed of 55 m.p.h. with its windows closed. We used the same 8 minute sample of noise for training and testing. To create a noisy sentence (approximately 10 seconds of speech), we selected a continuous block of noise from the long noise recording at random. This block of noise was scaled to achieve a given SNR level and added to the speech data. For these experiments, we computed the SNR on the unfiltered waveform, and designate this as SNR_wav.

Our main goal in this set of experiments was to compare the performance of the three proposed algorithms described in Section 2. However, to have a lower bound in the word error rate under noisy conditions, we also trained the genonic HMM recognizer from scratch using noisy training data at an SNR_wav of 6 dB. Therefore, we have two baseline recognizers, one based on "clean" HMMs and the other with "noisy" HMMs. The training data set consisted of 18,000 WSJ sentences from 170 male speakers. A compensation set was created using a subset of 300 sentences from the training set. The test set consisted of 90 sentences from 4 speakers.

Table 1 compares the performance for these systems. These results show that word error rate degrades from 11.1% for the clean/clean condition to 15.5% for the noisy/noisy condition. These baseline numbers will be used as a reference for the compensation algorithms.

	Test Clean	Test Noisy
Train Clean	11.1	22.2
Train Noisy	40.4	15.5

Table 1. Baseline word error rate in percent for clean and noisy conditions. The SNR_wav of the noisy data is 6 dB.

Table 2 compares the performance of the three compensation algorithms described in Section 2 and the baseline results.

Train	Test	Error Rate (%)
Clean	Clean	11.1
Noisy	Noisy	15.5
Clean	Noisy	22.2
Clean	Noisy+POF	18.2
Clean + Adaptation	Noisy	20.1
Clean + Adaptation	Noisy + POF	16.8

Table 2. Word error for baseline conditions and compensation algorithms. The SNR_wav of the noisy data is 6 dB.

We found that the error rate for mapping is 18.2% and for adaptation is 20.1%. In both cases we optimized each technique to maximize performance. For the combined approach, we found that adapting the HMM's to the mapped features reduced the error rate to 16.8%, only 8.4% ((16.8 - 15.5) / 15.5) worse than the full training in noise condition. Figure 1, which illustrates how the compensation algorithms perform at various SNRs, clearly shows how the combined approach outperforms mapping

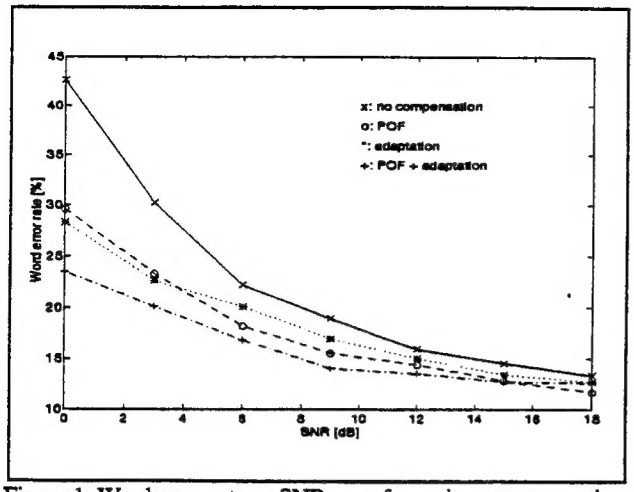

Figure 1: Word error rate vs SNR_wav for various compensation algorithms.

and adaptation at low SNR_wav levels. For SNR_wav levels above 15 dB, the POF-only approach produces the best performance.

All the previous experiments assume prior knowledge of the SNR level of the test data. This is not a serious assumption since we can always estimate the SNR at run-time and select the compensation models trained at a similar SNR. Table 3 shows

Model SNR_wav [dB]	Test SNR_wav levels [dB]							
	0	3	6	9	12	15	18	inf
0	23.5	21.4	21.7	24.0	29.2	36.1	47.3	90.6
3	25.1	20.1	17.2	17.2	19.6	22.0	25.7	71.0
6	26.6	20.8	16.8	15.0	15.1	16.5	18.2	45.8
9	28.7	22.0	17.5	14.0	13.2	12.8	14.1	30.1
12	30.7	22.7	18.2	13.9	13.5	12.8	12.6	21.9
15	32.4	23.4	18.7	14.8	13.2	12.7	12.7	17.4
18	36.4	25.1	19.2	16.0	13.6	12.4	12.6	14.7
inf	42.6	30.2	22.2	18.9	15.9	14.5	13.3	11.1

Table 3. Word error rate at various SNR_wav levels. Columns correspond to the test data SNR_wav and rows correspond to the SNR_wav used to compensate the clean models.

performance for the combined approach (mapping + adaptation) for the cases in which the testing SNR level may not match the compensation SNR level. This experiment shows that a precise estimate of the SNR is not necessary since performance seems to degrade slowly as the mismatch between the model SNR and the test data SNR increases.

In summary, front-end mapping and HMM adaptation can be combined to improve performance in a noisy channel at low SNR_wav levels. These conclusions are applied in the following section.

3.2. ARPA-Sponsored Benchmark Test (Spoke 10)

3.2.1. Development Test Results

This section describes the procedure used for the 1994 ARPA-sponsored CSR evaluation spoke 10 test. The test consisted of WSJ data (5,000-word vocabulary) corrupted with additive noise collected in three different cars. The car noise was recorded in an automobile traveling at 55 m.p.h. with all windows closed and the air-conditioning turned on, with an omnidirectional microphone clipped to the drivers' side sun visor. A one-minute sample of noise, preceding the noise segment added to the speech and scaled to each SNR level, is available for adaptation. Three noisy test sets were created using the same clean utterances and several different noise levels.

The results on the S10 development test set are shown below in Table 4. These experiments used a bigram language model on the male speaker subset (65 sentences) for car #1. The SNR's computed by NIST in the below table use an "A" frequency-weighted filter [13] before computing the SNR. Since car noise contains significant low frequency energies, applying a frequency weighted filter will shift the SNR levels compared to an unweighted SNR computation on the waveform (SNR_wav).

	Experimental Condition			
	1	2	3	4
POF Compensation	disabled	enabled	enabled	enabled
POF Feature		39-D Cep	13-D Cep+C0	13-D Cep+C0
POF # Gaussians		100	300	300
POF # Frames		3	5	5
Training Car Noises		1,2,3	1,2,3	1
Testing Condition (NIST SNR in dB)	**Word Error**	**Word Error**	**Word Error**	**Word Error**
12	80.6	48.9	47.5	43.2
18	53.2	29.8	29.0	26.5
24	29.6	20.7	20.7	18.7
30	19.0	15.9	18.1	15.8
inf	12.8			

Table 4. Word error rates for various conditions on the development test (car 1) set using a bigram language model.

The second line in Table 4 refers to what feature was used by the mapping. The # Gaussians and the # Frames are both parameters of the POF mapping algorithm. The fifth line in Table 4 indicates which car noises the algorithms were trained on: experiments 2 & 3 trained on all 3 car noises (which includes noise from the same car as the development test set), while experiment 4 only trains on a sample of noise collected from the development test set car. The word-error rate's are computed for each condition as a function of the A-weighted SNR.

From the results of Table 4, we see that:

- When we train using a sample of the testing noise we get better performance than when we train on multiple car noises.
- Mapping the full 39-dimensional cepstral vector (cep + first and second order derivatives) seems to perform better at higher SNR's than mapping only the cepstrum and computing the first and second derivatives on the mapped features.
- Condition 1 shows the performance with no compensation, and how the algorithms help more at higher SNR levels.

3.2.2. Evaluation Test Results

We trained many different POF mappings and HMM's, and selected the appropriate mapping at runtime. Using a one-minute sample of noise, we trained gender-dependent POF mappings for many different SNR levels. The gender selection was done using a Bayesian classifier trained with noisy data at a medium SNR level. The SNR was computed using the average of the log spectral SNR computed at the output of the filterbank in the signal processing stage. (This produced SNRs higher than the ones computed in Section 3.1., and is denoted SNR_spec).

To create the compensation models, the one-minute adaptation noise was added to a subset of the WSJ training data consisting of 300 waveforms with a variable scale creating gender and SNR-specific compensation data sets. The 300 waveform compensation sets were used to train both the mapping and the adaptation parameters. At low SNR_spec levels (9-24 dB), we used the combined method (POF + Adaptation), and at high SNR_spec levels (27-33 dB) we used the POF mapping alone. The results of this test are shown in Table 5. For the worst condition (Level 3) the ratio of the clean-speech error to the noisy-speech error was reduced roughly from 5 to 2 after applying the compensation algorithm.

Compensation	Clean	Level 1	Level 2	Level 3
Enabled	-	10.1	8.8	12.5
Disabled	7.1	18.7	11.5	35.0

Table 5. Word error rates for the 1994 ARPA-sponsored evaluation on the Spoke 10 test.

4. SUMMARY

This paper describes how to compensate HMM-based recognizers in the presence of steady additive noise. We compared performance of compensation algorithms that operate in the feature and model domains, and experimentally found that both approaches produced improved results over the baseline condition. A combination of mapping and adaptation, however, yielded the best results at low SNR levels.

ACKNOWLEDGMENTS

This research was partially supported by a grant, NSF IRI-9014829, from the National Science Foundation and by the Advanced Research Projects Agency through Office of Naval Research Contracts ONR N00014-92-C-0154.

REFERENCES

1. V. Digalakis and H. Murveit, "GENONES: Optimizing the Degree of Mixture Tying in a Large Vocabulary Hidden Markov Model Based Speech Recognizer," 1994 IEEE ICASSP, pp. I537-I540.
2. H. Murveit, J. Butzberger, V. Digalakis, and M. Weintraub, "Large-Vocabulary Dictation Using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques," 1993 IEEE ICASSP, pp. II319-II322.
3. H. Murveit, J. Butzberger, and M. Weintraub, "Performance of SRI's DECIPHER™ Speech Recognition System on DARPA's CSR Task," 1992 DARPA Speech and Natural Language Workshop Proceedings, pp. 410-414.
4. Linguistic Data Consortium, "ARPA Spoken Language Systems November 1994 CSR Hub and Spoke Benchmark Test Material," LDC CDROM Disk T8-1.1, file: ./et94spec.doc.
5. L. Neumeyer and M. Weintraub, "Probabilistic Optimum Filtering for Robust Speech Recognition," 1994 IEEE ICASSP, pp. I417-I420.
6. H. Gish, Y.L. Chow, and J.R. Rohlicek, "Probabilistic Vector Mapping of Noisy Speech Parameters for HMM Word Spotting," 1990 IEEE ICASSP, pp. 117-120.
7. A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition," Ph.D. Thesis, Carnegie-Mellon University, September 1990.
8. A. Erell and M. Weintraub, "Filterbank-Energy Estimation Using Mixture and Markov Models for Recognition of Noisy Speech," 1993 IEEE ASSP, vol. 1, no. 1, pp. 68-76.
9. M.J.F. Gales and S.J. Young, "HMM Recognition in Noise using Parallel Model Combination," 1993 Eurospeech, pp. 837-840.
10. J.A. Nolazco Flores and S.J. Young, "Continuous Speech Recognition in Noise using Spectral Subtraction and HMM Adaptation," 1994 IEEE ICASSP, pp. I409-I412.
11. V. Digalakis and L. Neumeyer, "Speaker Adaptation Using Combined Transformation and Bayesian Methods," submitted to 1995 IEEE ICASSP.
12. G. Doddington, "CSR Corpus Development," 1992 DARPA SLS Workshop, pp. 363-366.
13. K.D. Kryter, "The Effects of Noise on Man," 1985 Academic Press.